1982

# The Performance of the Collocation and Galerkin Methods with Hermite Bi-Cubics

W. R. Dyksen

Robert E. Lynch
*Purdue University*, rel@cs.purdue.edu

John R. Rice
*Purdue University*, jrr@cs.purdue.edu

Elias N. Houstis
*Purdue University*, enh@cs.purdue.edu

Report Number:
81-413

# THE PERFORMANCE OF THE COLLOCATION AND GALERKIN METHODS WITH HERMITE BI-CUBICS

*W.R. Dyksen, R.E. Lynch and J.R. Rice*

*Computer Science Department, Purdue University*

*E.N. Houstis*

*Computer Science Department, University of South Carolina*

*CSD-TR 413*

*September 3, 1982*

*ABSTRACT*

This paper presents a study of the performance of the collocation and Galerkin methods using Hermite bi-cubic basis functions. The linear systems generated by the two methods are solved by direct methods, band Gauss elimination or Cholesky factorization. The problem domain consists of linear, self-adjoint elliptic equations on two-dimensional rectangular domains. The measures of performance are computer time and memory needed to achieve moderate accuracy. An earlier study [Houstis et al. 1978] comparing finite element and finite difference methods observes that collocation uses less computer time than Galerkin. More recently, [Weiser et al. 1980] gave detailed operation counts which support this observation, but also gave substantial experimental evidence to the contrary. We use a new implementation of the collocation method by E.N. Houstis which is tailored for rectangular domains (the one used in [Houstis et al. 1978] was designed for general domains). We use the Galerkin implementation of Weiser et al.

We outline the process of comparing the performance of PDE software and discuss the difficulty of reaching definitive conclusions. We analyze the question of error measurement and note that the example given in [Weiser et al. 1980] as a counterexample to the practice of measuring the error at the grid points or knots (as done in [Houstis et al. 1978]) is also a counterexample to the practice recommended by Weiser et al. of measuring the error on a fixed set of points. We give an efficient and rigorous error measurement technique for non-singular problems.

This study strongly supports the hypothesis that (with these implementations of the methods) collocation performs better than Galerkin for both computer time and memory.

## 1. INTRODUCTION AND SUMMARY

This paper presents a study of the performance of two methods for solving linear, self-adjoint elliptic problems on two-dimensional rectangular domains. Both methods are finite element methods using Hermite bi-cubic basis functions and both use direct elimination for band matrices to solve the resulting systems of linear equations. The principal differences between the methods is in the discretization technique; one uses collocation and the other uses the Galerkin method. Given all of the above, there are still possible variations of these methods depending on just how the basis elements and equations are ordered. For the Galerkin (Rayleigh-Ritz) method one wants to preserve the symmetric positive definite property of the linear system, so there is less flexibility in the ordering. The ordering derived from the tensor product nature of the problem is the one used. There are several reasonable orderings for the collocation equations, see [Dyksen and Rice, 1982] for more information. We use the traditional ordering of the structural engineering community; to our knowledge it gives the best efficiency for band Gauss elimination. The methods and their implementations are described in more detail in Section 2.

Operation counts provide an easy, but fuzzy, comparison of methods. One assumes that the accuracy of two methods of the same order is the same and that the execution time in an implementation is proportional to the arithmetic in a simplified, asymptotic version of the method. When this approach is applied to the collocation and Galerkin methods, it indicates that the collocation method should execute faster. Detailed operation counts are given in [Weiser et al., 1980] (see tables 1 and 2) although they do not use these counts to make a detailed comparison of collocation and Galerkin for Hermite bi-cubics. Our interpretation of these counts is that, for moderate accuracy, collocation is likely to be more effective than Galerkin using Hermite bi-cubics. The operation counts approach has obvious shortcomings; the most obvious in the present context are:

1. The errors are not the same, Galerkin is usually more accurate
2. Coefficient and right side function evaluations are ignored. They dominate in many applications.
3. Simple variations (improvements) in an algorithm can dramatically change the actual amount of arithmetic done. See [Dyksen and Rice, 1982] for a specific example involving simple band Gauss elimination applied to the collocation equations.

The first systematic experimental data comparing collocation and Galerkin are those of [Houstis et al., 1978] which is a by-product of their comparison of the present collocation method with ordinary finite differences. The objective of the study of Houstis et al. was to show the superiority of high order finite elements methods over ordinary finite difference methods for solving elliptic problems on general domains. They observed that their collocation program was more efficient (when applied to rectangular problems) than their Galerkin program.

A second study of [Weiser et al., 1980] involves exactly the present problem area and five methods, including the collocation and Galerkin methods considered here. Weiser et al. claim to contradict the results of Houstis et al. and attribute the contrasting results to be due more efficient "assembly phase techniques" (i.e., in forming the equations to be solved). The results of Houstis et al. were based on programs designed for general domains because this was the problem area they studied. Thus the assembly phase of the Galerkin program

was substantially less efficient than that possible for programs tailored to rectangular domains.

We feel, however, that something was wrong with the Weiser et al. conclusions for the following reasons:

1. The work of the assembly phase is negligible for the simple problems used in their study.
2. The operations counts contained in their paper did not support their conclusions.

We thus prepared two collocation programs, INTERIOR COLLOCATION and HERMITE COLLOCATION, tailored to rectangular domains and compared them with the Weiser et al. program SPLINE GALERKIN implementing the Galerkin method. Our conclusions are stated below.

Weiser et al. also raise the question of how to measure the error of the computed solutions in an experimental study. The maximum error at the grid points is used by Houstis et al. as they were primarily involved with finite difference comparisons where there is no satisfactory method to measure the error off the grid points (especially for non-rectangular domains). Weiser et al. prefer measuring the error at some a priori fixed point set, specifically, on a 100 by 100 grid; this is easy for finite element approximations which are defined everywhere. It is, of course, well known that there exist problems where either scheme fails to provide an accurate error measurement. The merits of both approaches were discussed by Houstis et al. and they concluded that the differences would not be significant in any substantial statistical study. Weiser et al. present an example problem where measuring the error at the grid points gives completely unrealistic results. We analyze this example further (it has a subtle but strong pathological nature) and show that the measurement scheme preferred by Weiser et al. also gives completely unrealistic results for this example (they did not make the grid fine enough to see the effect). More significantly, we present an efficient method for measuring the error which gives a guaranteed upper bound for smooth problems (the example of Weiser et al. is highly singular).

## 2. THE METHODS AND THE SOFTWARE

The problem area is formulated mathematically as follows: We have a linear elliptic operator $L$, a rectangular domain R and wish to solve

$$L[u] = (p(x,y)u_x)_x + (q(x,y)\,u_y)_y + r(x,y)u = f \qquad (x,y) \in R \qquad (2.1a)$$

$$u = g \qquad (x,y) \in \partial R \qquad (2.1b)$$

where $f$ and $g$ are given functions. The Dirichlet boundary condition (2.1b) is a special case of *uncoupled boundary conditions*, that is, where

$$\begin{aligned} a(x,y)\,u + b(x,y)\,u_n &= g(x,y) \\ a(x,y)\,b(x,y) &\equiv 0 \\ a^2 + b^2 &> 0 \qquad \textit{all } (x,y) \in \partial R \end{aligned} \qquad (2.1c)$$

We approximate $u(x,y)$ by

$$U(x,y) = \sum_{i=1}^{N} a_i b_i(x,y)$$

where the $b_i(x,y)$ are the standard Hermite bi-cubic basis functions formed as a tensor product of the one dimensional Hermite cubics. The domain R is subdivided with a rectangular, tensor product grid into $n^2$ rectangles; the grid lines

are the knots of the Hermite bi-cubics. There are $N = 4(n+1)^2$ basis functions $b_i(x,y)$.

For the usual collocation method, the operator $L$ is expanded, a set of collocations points $(x_j,y_j)$ is chosen and (2.1) is approximated by

$$L[U](x_j,y_j) = f(x_j,y_j) \quad j = 1,2,\ldots,4n^2 \tag{2.2a}$$

$$U(x_j,y_j) = g(x_j,y_j) \quad j = 4n^2+1,\ldots,N \tag{2.2b}$$

The first $4n^2$ collocation points are placed at the four Gauss points of each subrectangle; this is known [Houstis, 1978], [Purcel and Wheeler, 1981] to give a fourth order discretization error for smooth problems. The remaining collocation points are distributed with two at the Gauss points of each grid segment on the boundary plus one at each of the four corners of $R$, see Figure 1. The basis functions are associated with the grid points, four per interior point, and are numbered from bottom to top, then left to right. If the problem (2.1) is homogeneous ($g(x,y)\equiv 0$), then the basis elements which are non-zero on $\partial R$ may be discarded (they are easily identified) which reduces $N$ from $4(n+1)^2$ to $4n^2$.

The ordering of the equations is that of the collocation points. The *finite element ordering* (traditional in structural engineering applications) is used. This ordering is not easy to express in algorithmic terms (it takes a dozen lines or so). The numbering given in Figure 1 is an example of this ordering and this pattern is used for larger values of $n$. A significant feature of this ordering is that, for uncoupled boundary conditions, the number of basis functions can be reduced as for the case of homogeneous boundary conditions. More significantly, this reduction does more than reduce $N$ from $4(n+1)^2$ to $4n^2$, it reduces the band width of the resulting linear system from $4n+11$ to $2n+5$.

The Galerkin equations for the same basis functions approximate (2.1a) with homogeneous boundary conditions by

$$\int_R L[U]b_i = \int_R f b_i \quad i=1,2,\ldots,N \tag{2.3a}$$

Since $L$ is self-adjoint, Green's theorem can be applied to (2.3a) to obtain the more common form

$$\sum_{j=1}^{N} \int_R (p b_{jx} b_{ix} + q b_{jy} b_{iy} + r b_i b_j) a_i = \int_R f b_i \quad i=1,2,\ldots,N \tag{2.4a}$$

As in the case of the collocation method, the number of basis functions may be reduced from $4(n+1)^2$ to $4n^2$ for a homogeneous problem. This reduction is not made by the software used in this study; the reduction is only of modest benefit for larger values of $n$. If (2.1) is not homogeneous then the boundary conditions are satisfied by a penalty function method.

More details of these methods are given in [Weiser et al., 1980]. Each method has three distinct steps:

1. Discretization: selection of basis functions and approximations to the continuous problem (2.1).
2. Indexing: choice of ordering the equations and unknowns.
3. Solution of a linear algebraic system of equations.

We evaluate the performance of these methods by using two specific implementations from the ELLPACK system [Rice, 1981]. They are SPLINE GALERKIN (DEGREE=3, SMOOTH=1) written by A. Weiser (and used for the study [Weiser et al., 1980]) and INTERIOR COLLOCATION written by E. Houstis. INTERIOR COLLOCATION applies only to uncoupled boundary conditions. It uses the fact that the Hermite bi-cubics are the dual basis to value and derivative evaluation to

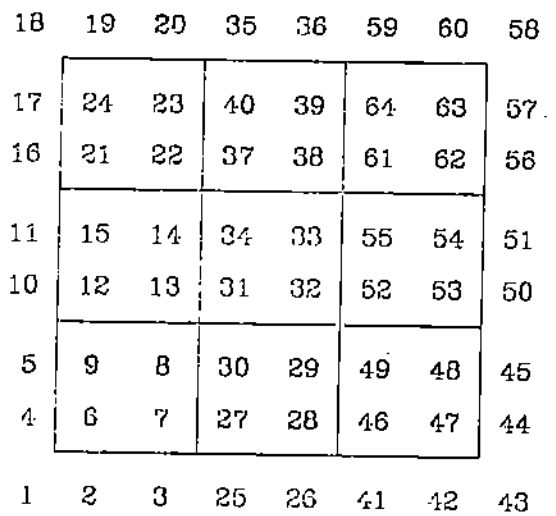



Figure 1. The collocation points for $n=3$. The numbers are at the location of the collocation points and they indicate the ordering of the equations used. All points are at the Gauss points of their respective domains.

precalculate the coefficients of $8n+4$ basis functions associated with the boundary making $U(x,y)$ satisfy the uncoupled boundary condition (2.1c) *without* collocating on the boundary.

In the analysis of measuring the error, we also use HERMITE COLLOCATION written by E. Houstis which handles general linear boundary conditions. Both collocation programs are specifically designed for rectangular domains and are not the ELLPACK program called COLLOCATION as used for general domains in the study [Houstis et al., 1978]. They compute the same approximation when applied to a problem with uncoupled boundary conditions.

In principle, both the collocation and Galerkin methods can take advantage of homogeneous boundary conditions to reduce the number of unknowns in the problem. The advantage is, at first glance, worthwhile, but not large: it reduces the size of the problem by a factor of $(1-2/n)$ for large $n$. Data given later support the assumption that the reduction in the number of unknowns is not very important for large problems. However, there is a much more dramatic effect in the case of INTERIOR COLLOCATION where dropping the uncoupled boundary condition equations also halves the band width of the resulting linear system. Thus, INTERIOR COLLOCATION takes advantage of homogeneous boundary conditions (and more) while SPLINE GALERKIN does not.

Figure 2 shows the pattern of non-zero elements in the linear system of equations generated for the Laplacian; both the usual and the interior collocation patterns are shown for collocation. The Galerkin matrix is symmetric, positive definite with at most 36 non-zero elements in each row and with bandwidth about $6n$. The collocation matrix is non-symmetric with at most 16 non-zero elements in each row. Its bandwidth (using the finite element ordering) for uncoupled boundary conditions is about $2n$, otherwise it is about $4n$. These equations are solved by the programs LINPACK SPD (the LINPACK implementations of Cholesky factorization of symmetric positive definite matrices) and BAND GE (ELLPACK implementation of LINPACK's Gauss elimination for band matrices modified to do scaled partial pivoting).

Figure 3 shows the patterns of non-zeros for two orderings of the Galerkin equation for $n=4$ (100 equations). The tensor product ordering is the one used by SPLINE GALERKIN. Note that the finite element ordering gives a smaller band width.

## 3. PERFORMANCE EVALUATION

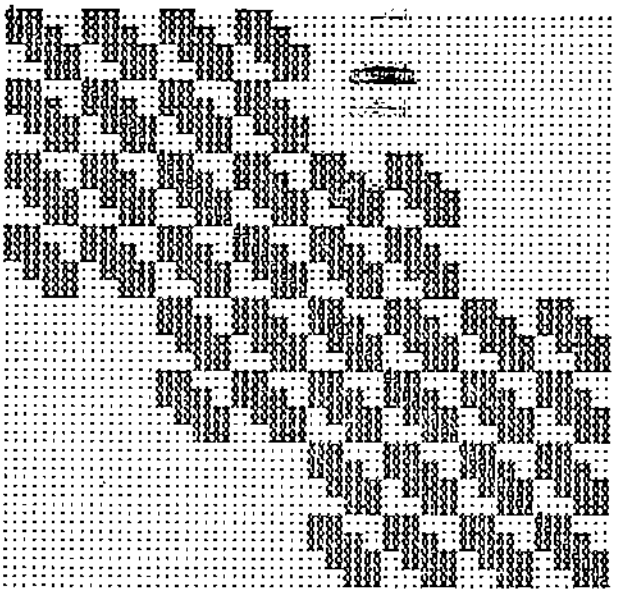
Studies to evaluate the performance of numerical methods are not easy to make. They also have a history of being done rather casually, see [Crowder et al., 1979]. We follow the methodology of [Rice, 1979a] and [Houstis and Rice, 1980] using the system designed for this purpose, [Boisvert et al., 1979]. A performance evaluation can be invalidated by one error (in design or technique) in any one of several places. Once one concedes that the design and technique are correct, there remain two fundamental questions:

*What is a numerical method?*

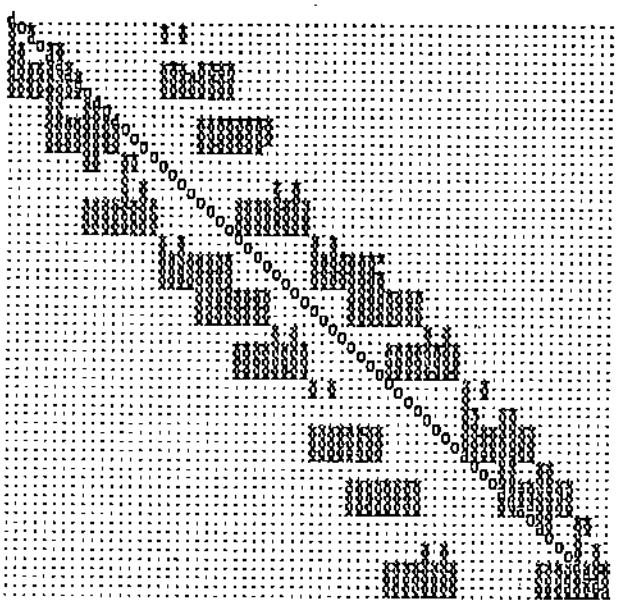
*To what set of problems does the performance evaluation apply?*

These questions are addressed in some detail in the references mentioned above. There are two principal facts:

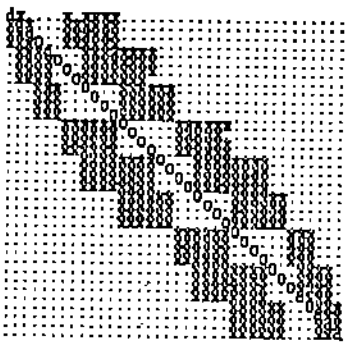
(a) *Numerical methods are ambiguously defined;* the apparent precision of textbook descriptions melts into great uncertainty in actual computations. One does not evaluate methods; one evaluates specific implementations of



A



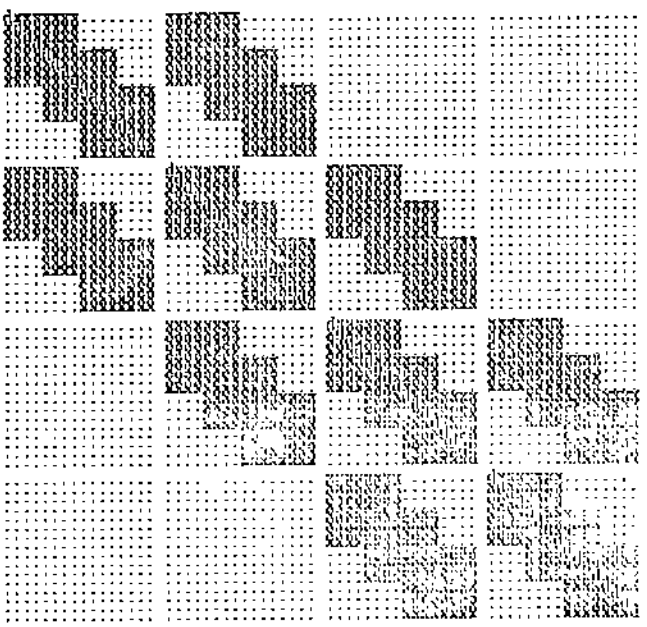
B



C

Figure 2. The patterns of non-zero elements for $n=2$ in (A) the Galerkin matrix, (B) the collocation matrix and (C) the collocation matrix with uncoupled boundary condition equations removed.

A



B

Figure 3. The patterns of non-zero elements for $n=2$ in the Galerkin equations for (A) the tensor product ordering and for $n=2$ (B) the finite element ordering.

methods, that is, computer programs. Even with specific computer programs there are uncertainties introduced by compilers, operating systems and computer hardware.

In this study we evaluate the performance of the programs SPLINE GALERKIN (DEGREE=3, SMOOTH=1) + LINPACK SPD and INTERIOR COLLOCATION + BAND GE. These implementations are within the class used in the earlier studies; the programs are all variants of the programs used in earlier studies. We believe that there might be other implementation techniques for the collocation and Galerkin methods which are superior to these.

(b) *The population of problems to which the numerical methods are to be applied is unknown.* One only has the vaguest sort of knowledge about the elliptic problems that occur in practice. The mathematical definitions of problem populations are precise (e.g., $u(x,y) \varepsilon C^1(R)$) but clearly irrelevant. The subset of elliptic problems with $u \varepsilon C^4$ which have fifth order derivatives nowhere continuous is of measure 1 while in practice there are no such problems.

The only approach currently known to define the subject population is by enumerations of a set of parameterized problems. Such a set is given by [Rice et al., 1981] and we use a subset of 18 problems. Their numbers are:

1-1, 3-1, 4-1, 5-1, 5-4, 6-1, 8-2, 9-1, 10-2,
10-3, 11-2, 17-2, 22-1, 33-1,41-3, 47-2, 50-1, 54-2

These 18 problems represent 10 different elliptic operators; 9 problems have homogeneous boundary conditions. All the problems are listed in Appendix 2. This problem set is intended to represent the simple to moderately complex problems that arise in practice.

The study of [Houstis et al., 1978] used 6 problems with 4 different operators: 1-1, 3-2, 4-1, 5-6, 6-1, 10-3. Problem 3-2 has parameter $\alpha$=2.5 while 3-1 has parameter $\alpha$=1.5. The study of Weiser et al. used 13 problems with 5 different operators: 1-1, 3-1, 3 (with $\alpha$=2.25), 5-7, 6-1, 7-1, 10 (with $\alpha$=10, $\beta$=.3), 10 (with $\alpha$=100, $\beta$=.3), 25-1, 25-2, 25 (with $\alpha$=2), 25 (with $\alpha$=3). The present study is based on a larger and considerably more varied problem set than the two previous studies.

Performance is evaluated by the accuracy achieved as a function of computer time and memory used. The accuracy is measured as the maximum error divided by the maximum value of $u(x,y)$; see the next section for a complete discussion on the estimation of accuracy. The time and memory used are measured on a VAX 11/780 computer with floating point accelerator using the UNIX Fortran compiler f77. See [Rice, 1982] for a discussion of the probable variations of relative computer time as a function of machine and Fortran dependencies. We expect the variations to be smaller than the "normal" 20-40 percent because the computations done by these programs are very similar in nature.

The statistical methodology used is a simple non-parametric analysis. One ranks the two methods on each problem and computes the average rank. One then obtains confidence intervals on the observed differences, see [Hollander and Wolfe, 1973] for details. The principal purpose of these statistics is to ensure that one has taken a large enough population so that the observed results are not due to chance.

## 4. ERROR MEASUREMENT

We discuss three topics in this section:

1. The measurement of error on finite point sets and pathological functions that give misleading results.
2. A reliable and efficient method of estimating the error for well-behaved problems.
3. Is there faster convergence at the grid points or, equivalently, it is inherently less reliable to estimate the error using only grid point values.

It is well known that it is unreliable to estimate the accuracy by measuring the error on a finite point set. Given any two finite point sets A and B, it is easy to construct elliptic problems where A gives reliable error estimates and B does not. Such constructions usually involve "pathological" problems and hence most people feel comfortable estimating the error on a finite point set provided it is of reasonable size.

Weiser et al. considered the problem

$$\begin{aligned} (x^2u_x)_x + (y^2u_y)_y - (xy)^2u &= f \quad R=\{0<x,y<1\} \\ u &= 0 \quad \text{on } \partial R \end{aligned} \tag{4.1}$$

The right side $f(x,y)$ is chosen to make the solution $u(x,y) = e^{x+y}(x^2-x)(y^2-y)$ which is an entire function. Note that this problem is very degenerate at the origin, the elliptic equation reduces to 0=0 with boundary conditions zero.

Weiser et al. note that measuring the error of collocation at the grid points is not reliable. They then conclude that it is inherently unreliable to estimate the error using the grid points and they recommend measuring the error on an a priori, fixed set of points. Weiser et al. state that collocation "seems to be making large errors in approximating the normal derivative across the domain boundaries x=0 and y=0".

In fact, the situation is quite different and (4.1) is a very special, pathological problem where *both* methods of estimating the error are unreliable. A contour plot of the error in collocation is given in Figure 4; note that the error consists of one bump inside the grid square at the origin. This situation is independent of $n$ as shown below. Thus when $n$ is large enough the support of the bump will miss any fixed set of points and render unreliable the error estimation technique advocated by Weiser et al. However, this example only illustrates what we already know; there is no generally reliable way to estimate the error using a finite set of points. As explained below, if the exponent 2 in the coefficient of u is changed to any other number, this pathological error behavior of collocation disappears completely.

An examination of the graphs of error versus time of Weiser et al., shows that, with one exception, the Galerkin and collocation errors are the same order of magnitude. The single exception, (4.1), shows nearly constant 100% error for collocation. This is because the coefficient of one of the basis elements associated with the mesh square S with vertex the origin has nothing to do with the differential equation problem. In fact, the plot of the error shown in Figure 4 essentially gives a contour plot of this basis element. Furthermore, the system of linear equations is nearly singular in the sense that a small change in the differential equation makes the system singular.

The differential operator for (4.1) is

$$L[u] = M[u] - \alpha x^2y^2u, \quad M[u] = (x^2u_x)_x + (y^2u_y)_y,$$

with $\alpha=1$. The bi-cubic Hermite approximation to the solution is

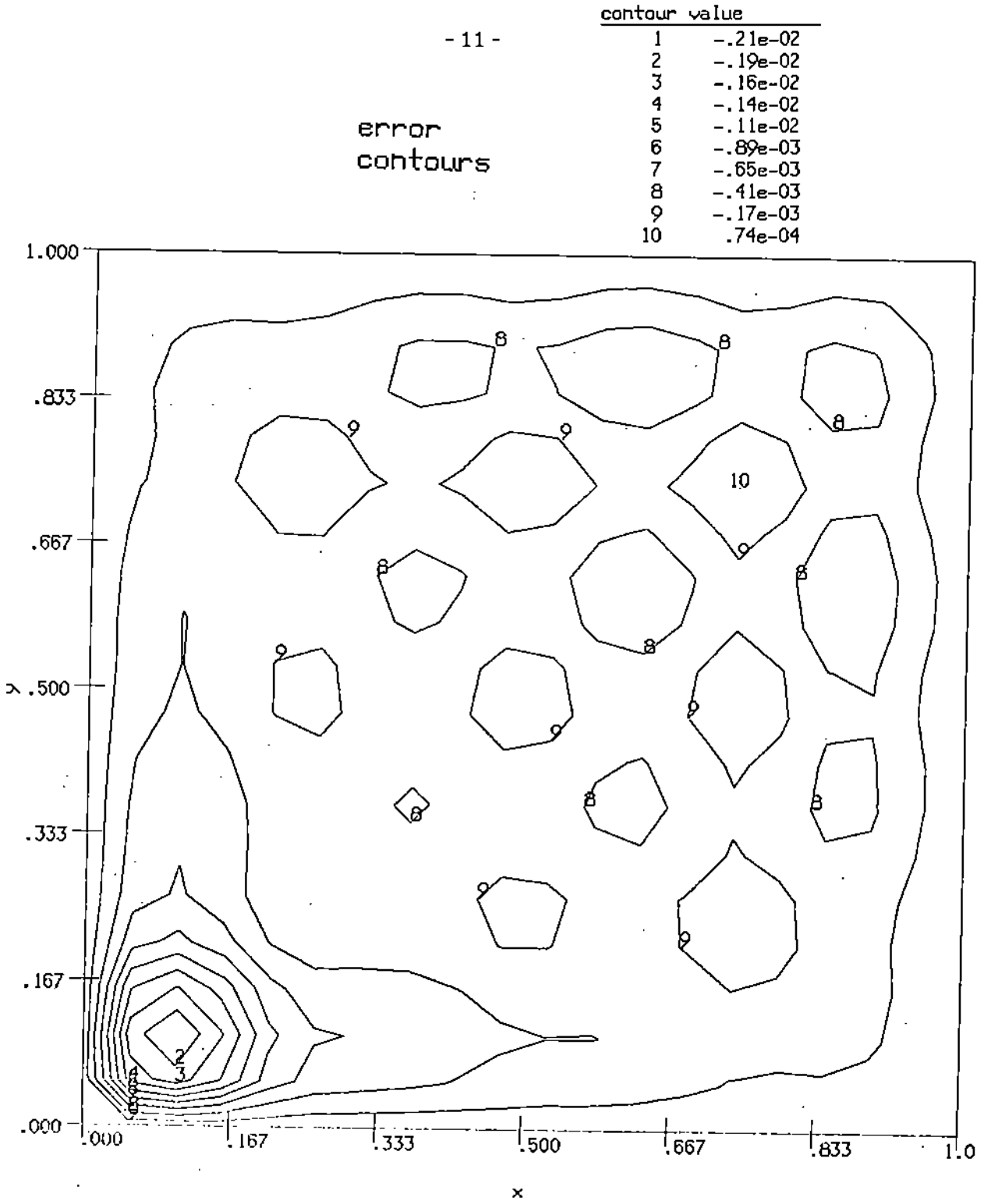



Figure 4. A contour plot of the error in collocation for the problem (4.1) for $n=4$. The bump in the error persists in the lower left grid square as $n$ increases.

$$\sum_{k,l=0}^{n-1} \sum_{i,j=1}^{2} \alpha_{i,j,k,l} s_i(x - kh) s_j(y - kh)$$

where

$$s_0(z) = 2(z + h/2)(z - h^2)^2/h^3, \quad s_0(-z) = s_0(z), \quad 0 \le z \le h,$$
$$s_1(z) = z(z - h^2)/h^2, \quad s_1(-z) = -s_1(z), \quad 0 \le z \le h,$$
$$s_0(z) = s_0(-z) = s_1(z) = s_1(-z), \quad h < z.$$

The differential operator $L$ is applied to the approximation and it is then evaluated at the collocation points. Each evaluation gives the left side of one of the equations in the linear system.

Since

$$(z^2 s_1'(z))' = 12z(z - \tau_1)(z - \tau_2), \quad \tau_p = h(1 \pm \sqrt{1/3})/2, \ p = 1, 2,$$

$(z^2 s_1'(z))'$ is zero at the collocation points $\tau_1$ and $\tau_2$. Therefore, at the four collocation points $(\tau_p, \tau_q)$, $p, q = 1, 2$, in the mesh square S, we have

$$K_{p,q} \equiv L[s_1(x)s_1(y)]_{(\tau_p,\tau_q)} = a{\tau_p}^2{\tau_q}^2 s_1(\tau_p)s_1(\tau_q)$$

because the operator $M$ applied to $s_1(x)s_2(y)$ is zero that these four points and thus the effect of the derivatives is not modelled. Furthermore, $K_{p,q}$ is $0(h^6)$.

The values of $K_{p,q}$ are the coefficients of the unknown $\alpha_{1,1,0,0}$ in the four equations of the linear system in which that unknown appears; the coefficients of all the other unknowns $\alpha_{i,j,k,l}$ are $O(h^{i+j})$, $i,j = 1,2$. It follows from Cramer's Rule that the the value of $\alpha_{1,1,0,0}$ is orders of magnitude larger than the other coefficients.

*Error in bi-cubic approximation.* Next we show that the difference in the max-norm of the errors of bi-cubic approximation schemes, such as Galerkin and collocation, can be determined by evaluation at about $9m$ points where $m$ is the number of mesh squares. Throughout this discussion, we use $R = [-h,h] \times [-h,h]$ and the points

$$x_1 = y_1 = -h, \ x_2 = y_2 = -\vartheta h, \ x_3 = y_3 = \vartheta h, \ x_4 = y_4 = h,$$

where $\vartheta$ is a parameter, $0 < \vartheta < 1$. We also define $K$ by

$$h^4 K = \max_{-h \le x \le h} (x^2 - h^2)(x^2 - \vartheta^2 h^2)/48.$$

We begin by determining a bound on the error $e = u - p$ on R, where $p$ is the bi-cubic interpolant to a given function $u$ defined by

$$p(x_j, y_k) = u(x_j, y_k), \quad j, k = 1, 2, 3, 4. \tag{4.2}$$

The bi-cubic $p$ can be written in the Lagrange form of the interpolation polynomial as

$$p(x,y) = \sum_{k=1}^{4} \sum_{j=1}^{4} u(x_j, y_k) l_j(x) l_k(y),$$

where

$$l_j(x) = \prod_{\substack{k=1 \\ k \ne j}}^{4} \frac{(x - x_k)}{(x_j - x_k)}.$$

We use $L$ to denote the Lebesgue constant for this basis:

$$L = \max_{-h \le x \le h} \sum_{j=1}^{4} |l_j(x)|. \tag{4.3}$$

Because cubic interpolation to a constant function is exact, we have

$$\sum_{j=1}^{4} |l_j(x)| = \begin{cases} 1 - 2l_3(x) & -h \le x \le -\vartheta h, \\ 1 - 2l_1(x) - 2l_4(x) & -\vartheta h \le x \le \vartheta h, \\ 1 - 2l_2(x) & \vartheta h \le x \le h, \end{cases} \tag{4.4}$$

which is a piecewise cubic polynomial. In each of the three subintervals, in (4.4), the global maximum occurs at an interior point, hence $L$ can be determined by finding zeros of three quadratic polynomials.

THEOREM 1. *On* $R = [-h,h]^2$ *the error* $e = u - p$ *of the bi-cubic interpolant* (4.2) *satisfies*

$$|e(x,y)| \le h^4 K(1 + L)[\|u_{xxxx}\|_\infty + \|u_{yyyy}\|_\infty]. \tag{4.5}$$

*Proof.* The error as a function of $y$ along the four lines $x = x_j$ is given by the Cauchy form of the interpolation error as

$$e(x_j,y) = (y^2 - h^2)(y^2 - \vartheta^2 h^2)u_{yyyy}(x_j,\xi)/24. \tag{4.6}$$

For each $y \in [-h,h]$, let $q(\cdot,y)$ be the cubic polynomial in $x$ which interpolates to $e(\cdot,y)$ at $x = x_j$, $j = 1, 2, 3, 4$; then

$$q(x,y) = e(x_1,y)l_1(x) + e(x_2,y)l_2(x) + e(x_3,y)l_3(x) + e(x_4,y)l_4(x). \tag{4.7}$$

Thus with $L$ the Lebesgue constant (4.3), we obtain from (4.6) that

$$|q(x,y)| \le L \max_j |e(x_j,y)| \le 2h^4 KL\|u_{xxxx}\|_\infty. \tag{4.8}$$

By (4.7), $q(\cdot,y)$ is the cubic interpolant in $x$ to $e(\cdot,y)$, and therefore

$$\begin{aligned} q(x,y) &= e(x,y) + (y^2 - h^2)(y^2 - \vartheta^2 h^2)e_{xxxx}(\eta,y)/24 \\ &= e(x,y) + (y^2 - h^2)(y^2 - \vartheta^2 h^2)u_{xxxx}(\eta,y)/24, \end{aligned}$$

where the last equality holds because $e_{xxxx} = u_{xxxx} - p_{xxxx}$ and $p$ is a cubic polynomial in $x$. Consequently we have

$$|e(x,y)| \le |q(x,y)| + 2h^4 K\|u_{xxxx}\|_\infty. \tag{4.9}$$

Combining (4.8) and (4.9), interchanging the roles of $x$ and $y$, and averaging, we obtain (4.5).

We now obtain a bound on the error $E = u - t$, where $t$ is an arbitrary bi-cubic polynomial on the mesh-square R. We set $d = p - t$ and use the notation

$$\|d\|_R = \max_{j,k=1,2,3,4} |d(x_j,y_k)|.$$

THEOREM 2. *On* R *let* $t$ *denote a bi-cubic polynomial and* $p$ *the bi-cubic interpolant to* $u$*; set* $d = p - t$. *Then the error* $E = u - t$ *satisfies*

$$|E(x,y)| \le L^2\|d\|_R + h^4 K(1 + L)[\|u_{xxxx}\|_\infty + \|u_{yyyy}\|_\infty]. \tag{4.10}$$

*Proof.* The difference $d$ is given by

$$d(x,y) = \sum_{k=1}^{4} \Big[ \sum_{j=1}^{4} d(x_j,y_k)l_j(x) \Big] l_k(y)$$

and thus

$$|d(x,y)| \le L^2\|d\|_R. \tag{4.11}$$

Because $u = t + E = p + e$, $|E(x,y)| \le |d(x,y)| + |e(x,y)|$ and so (4.10) follows from (4.11) and Theorem 1 which concludes the proof.

*Remark.* Since the discretization error of collocation is $O(h^4)$, the right side of (4.10) is the sum of two $O(h^4)$ terms. If interpolation of higher order $p$ is used then the same argument shows that the right side can be replaced by

$$L^2\|d\|_R + O(h^{p+1})$$

This allows one to compute precisely, for small $h$, a bound on the error.

Theorem 2 provides an efficient and reliable method to measure the error of a bi-cubic Hermite approximation for a problem with smooth solution. If $u$ is defined on $\bar{\Omega}$ which is the union of $m$ mesh-squares of side length h, then Theorem 2 can be applied to each mesh-square. If the bi-cubic approximation $t$ is also continuous, then one needs a total of about $9m$ evaluations of the error $E = u - t$ because along some of the mesh-square edges, the evaluation points are shared by two adjacent mesh-squares.

One gets an $O(h^4)$ estimate of the global maximum error as $L^2$ times the maximum of the difference $d$ at all the evaluation points. If one also computes upper bounds on the maxima of the fourth derivatives of the function $u$, then one can determine an upper bound on the global error $E$.

COROLLARY 1. *Let $g$ and $c$ denote the Galerkin and collocation bi-cubic approximations to the solution $u \in C^4$ of an elliptic partial differential equation problem. The difference between the maximum error $u - g$ and $u - c$ is bounded by the maximum of $L^2\|g - c\|_R$ over all the mesh squares.*

*Proof.* One can write the second error as $(u - g) + (g - c)$ and the result is a direct consequence of Theorem 2.

Superconvergence is a phenomenon of some finite element methods based on higher order splines where the observed error at the grid points (knots) is of higher order than the global error. The dominant error term is zero at the grid points so the error is governed by a second, higher order term. For two dimensional problems the two terms involved are of the same order, so superconvergence is not expected. However, there might be something special about the grid points and in [Houstis et al., 1978] it was observed that the error at the grid points is smaller for 4 of 17 problems by a factor two to four. This phenomenon was not observed for the Galerkin or least squares methods.

We have used error estimates based on the grid points in performance evaluation for several reasons. The most important is that, for finite difference methods, this is the only measurable error. Other reasons are (i) a general feeling that the error estimated with the grid points does not differ much from other error estimates, (ii) the opinion that a change of 50 to 100 percent in the error should not affect most performance evaluations. That is, if doubling the error affects the outcome of a comparison, then the methods are probably reasonably equivalent since there are other, uncontrollable and equally large uncertainties in the evaluation. Note that the fixed perturbations in the error are less important for high order methods (such as the two studied here) than for low order methods. These reasons were not, however, based on any systematic analysis.

We have collected data on error estimates and give histograms for the ratio

$$\frac{\text{Error estimated with a 51 by 51 set of points}}{\text{Error estimated with the grid points}} \qquad (4.12)$$

Figure 5 gives the histograms are for $n$=8, 12 and $n$=28 and for all 69 problems which have been solved by HERMITE COLLOCATION. Figure 6 gives the histograms of (4.12) for both collocation and Galerkin using $n$=8,28 and all 18 problems of this study.

We observe the following: For a coarse mesh (e.g., $n$=8), the error ratio (4.12) is substantially larger than 1 for 30 to 40 percent of all problems. For a fine mesh (e.g., $n$=28) only a few problems give ratios larger than 2; we believe some of these represent true extra accuracy at the grid points and some represent accidents of where the error is measured.

For the 18 problems in this study there is more of a spread than for the larger collection of 69 problems in Figure 5A. The data for these problems has

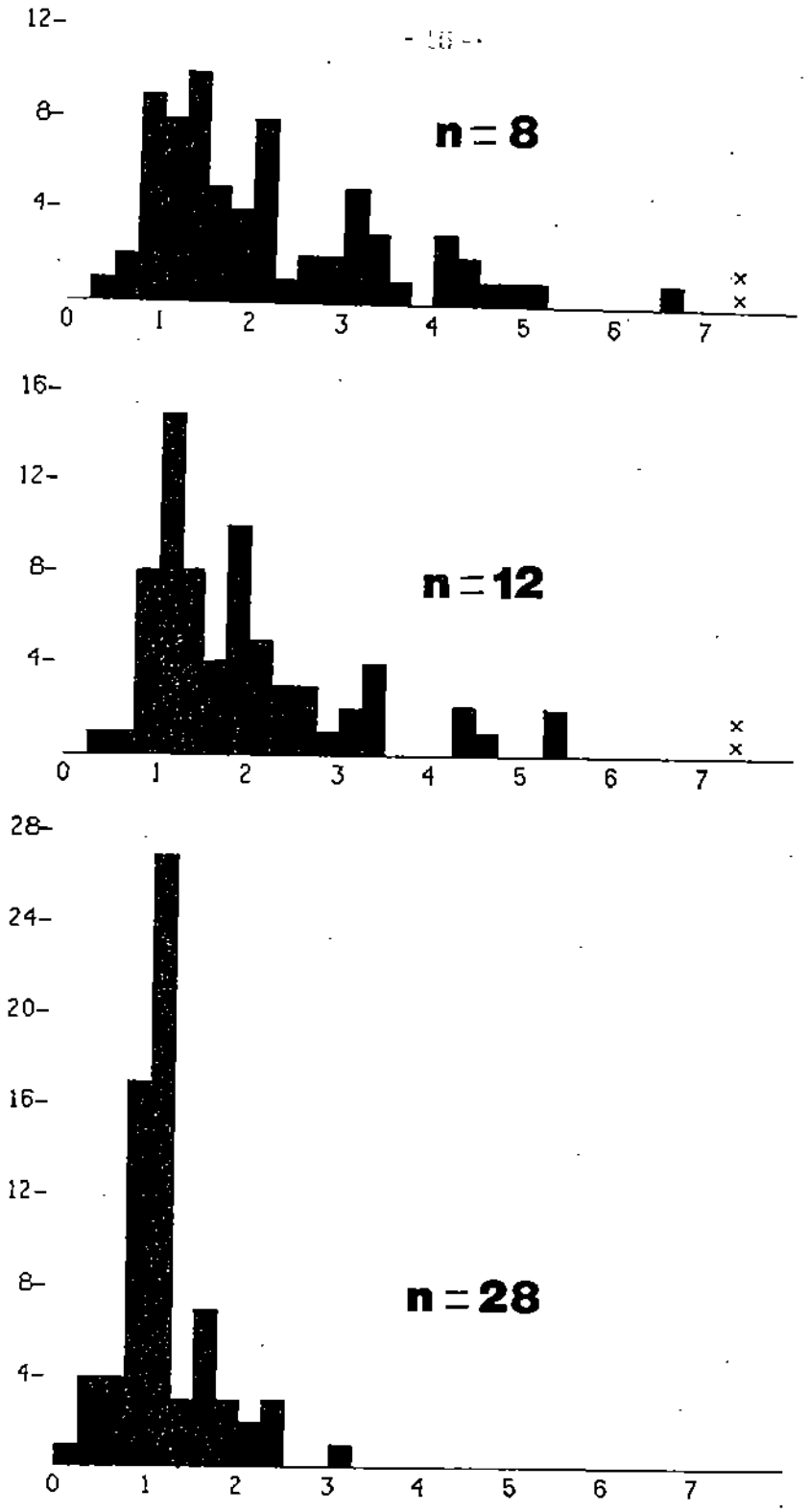



Figure 5. Histograms of the error ratio (4.12) using $n$=8, 12 and 28 for all 69 problems solved by HERMITE COLLOCATION. The x's represent points off the scale.

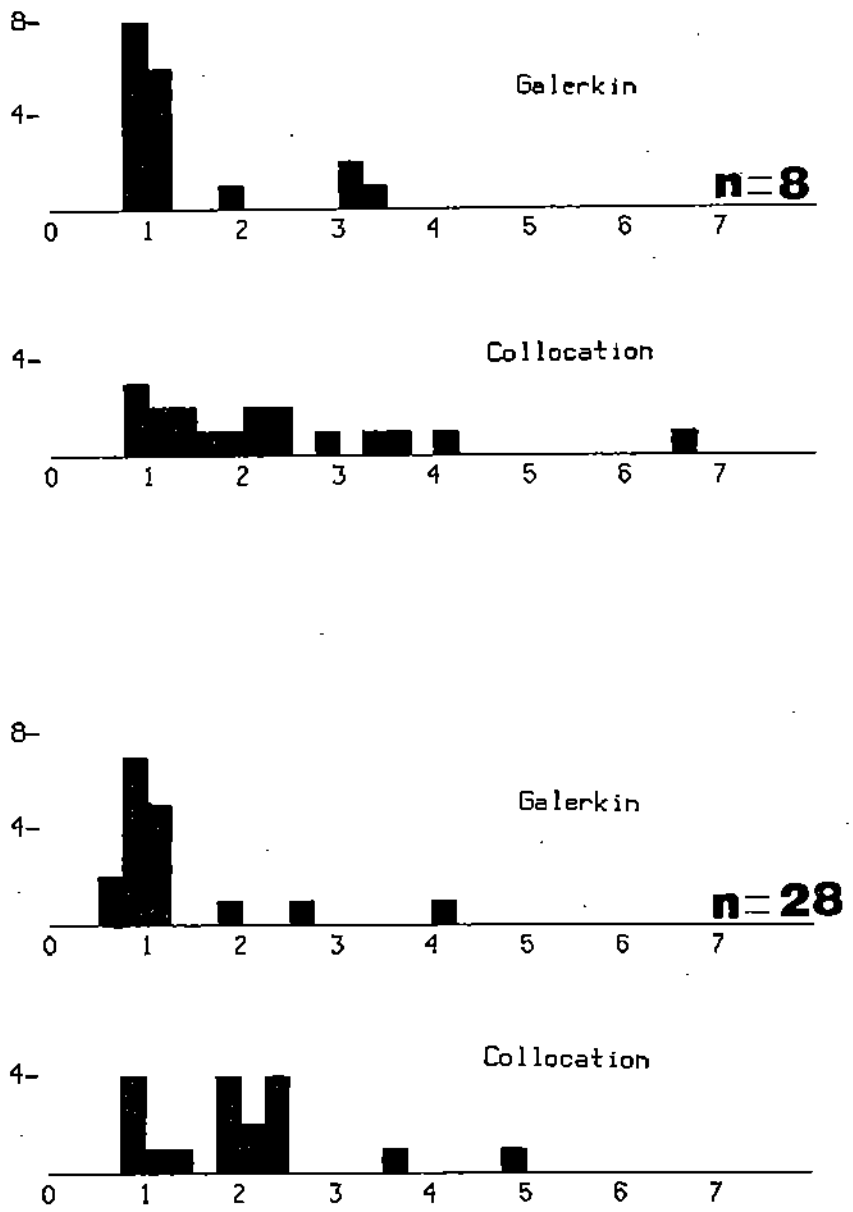



Figure 6. Histograms (for both collocation and Galerkin methods) of the error ratio (4.12) using $n=8$ and 28 for all the 18 problems of this study.

been examined in detail and a subjective judgement is that substantial "extra accuracy" occurs at the grid points for nine problems: 4-1, 5-4, 9-1, 11-2, 22-1, 33-1, 41-3, 47-2 and 50-1. There is no obvious characteristic shared by these problems except they are self-adjoint. It might well be that extra accuracy occurs more frequently for self-adjoint problems; small or moderate amounts of extra accuracy were judged to be present for five other problems.

We made the same examination for the Galerkin method and judged that some extra accuracy at the grid points occurs for problems 3-1, 6-1 and 41-3. The extra accuracy was similar to that observed for collocation for problems 3-1 and 41-3 while collocation exhibits the opposite effect for problem 6-1. Significantly, there was no special behavior in the accuracy at the grid points for any of the problems with non-homogeneous boundary conditions. This suggests that the least squares penalty function method used to satisfy the boundary conditions destroys whatever it is that makes the grid points special.

## 5. PERFORMANCE ANALYSIS

The 18 problems were solved by the two methods using the system [Boisvert et al, 1979] based on ELLPACK to assist such studies. The results are evaluated on the following criteria of performance:

1 Slope of error versus computer time

2 Time to achieve 3 significant digits of accuracy

3 Memory requirements

The memory criterion is the simplest, so we deal with it first. The principal use of memory should be the space used to solve the linear system of equations; at least for n reasonable large. Asymptotically the size of these spaces are:

$24n^3$ for Galerkin (SPLINE GALERKIN)
$48n^3$ for collocation (HERMITE COLLOCATION)
$24n^3$ for collocation and uncoupled boundary conditions (INTERIOR COLLOCATION)

These asymptotic estimates are well correlated with the measured memory used except for Galerkin; the SPLINE GALERKIN (DEGREE = 3, SMOOTH) + LINPACK SPD software uses about twice as much memory as one expects. We believe this is due to making an extra copy of the matrix as part of putting the software into the ELLPACK system. We observe in this study that INTERIOR COLLOCATION and SPLINE GALERKIN use about the same memory while HERMITE COLLOCATION uses about 75% more.

The ranks of the two methods using the first two criteria are given in Table 1. Ranks based on estimating the error at a fixed, 20 by 20 mesh and at the grid points are given. When the performances are nearly equal (less than 5% difference) both methods are ranked 1 (highest). We see that there is a substantial difference in the ranks depending on where the maximum error is measured. With the error measured at the grid points, collocation is clearly the better in both performance criteria. The average ranks and confidence levels are summarized in Table 2. An average rank of 1.00 means the method is always best in that performance measure; 2.00 means it is always worst. For example, in the case of 3 digits of accuracy measured at grid points, the rank of collocation is 1.06 and of Galerkin is 1.78. This difference in average ranks is significant at the 99% level of confidence. We also compared the performance on the basis of the least squares error at the grid points; the rankings are identical with those of the maximum error at the grid points.

*TABLE 1:* Ranks of INTERIOR COLLOCATION (COL) and SPLINE GALERKIN (GAL) using performance criteria 1 (slope) and 2 (3 digits).

| | Error at grid points | | | | Error on 20×20 mesh | | | |
|---|---|---|---|---|---|---|---|---|
| | Slope | | 3 digits | | Slope | | 3 digits | |
| Problem | COL | GAL | COL | GAL | COL | GAL | COL | GAL |
| 1-1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 2 |
| 3-1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| 4-1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 2 |
| 5-1 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 1 |
| 5-4 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 1 |
| 6-1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 |
| 8-2 | 1 | 1 | 2 | 1 | 1 | 2 | 2 | 1 |
| 9-1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 |
| 10-2 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 1 |
| 10-3 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 1 |
| 11-2 | 1 | 2 | 1 | 1 | 1 | 2 | 1 | 1 |
| 17-2 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 |
| 22-1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 |
| 33-1 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 |
| 41-3 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 1 |
| 47-2 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 |
| 50-1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 54-2 | 1 | 1 | 2 | 1 | 1 | 1 | 2 | 1 |
| Average for 18 problems | 1.06 | 1.44 | 1.17 | 1.67 | 1.06 | 1.28 | 1.16 | 1.22 |

*TABLE 2:* Summary of method ranks and significance. The significance entry is the confidence level at which the difference in average ranks is statistically significant. The average ranks do not sum to 3 because of ties.

Error estimated at grid points

| | Collocation | Galerkin | Significance |
|---|---|---|---|
| slope of error | 1.06 | 1.44 | 90% |
| 3 digits accuracy | 1.06 | 1.78 | 99% |

Error estimated at 20×20 mesh

| | Collocation | Galerkin | Significance |
|---|---|---|---|
| slope of error | 1.06 | 1.28 | none |
| 3 digits accuracy | 1.16 | 1.22 | none |

Four typical performance plots of computer time versus error are shown in Figure 7. The scales are logarithmic and values are plotted for $n$=4, 8, 12, 20 and 28. Problem 33-1 has collocation performance noticeably better at the grid points and the two methods are about the same on a 50×50 mesh. Problem 1-1 has collocation performance better both at the grid points

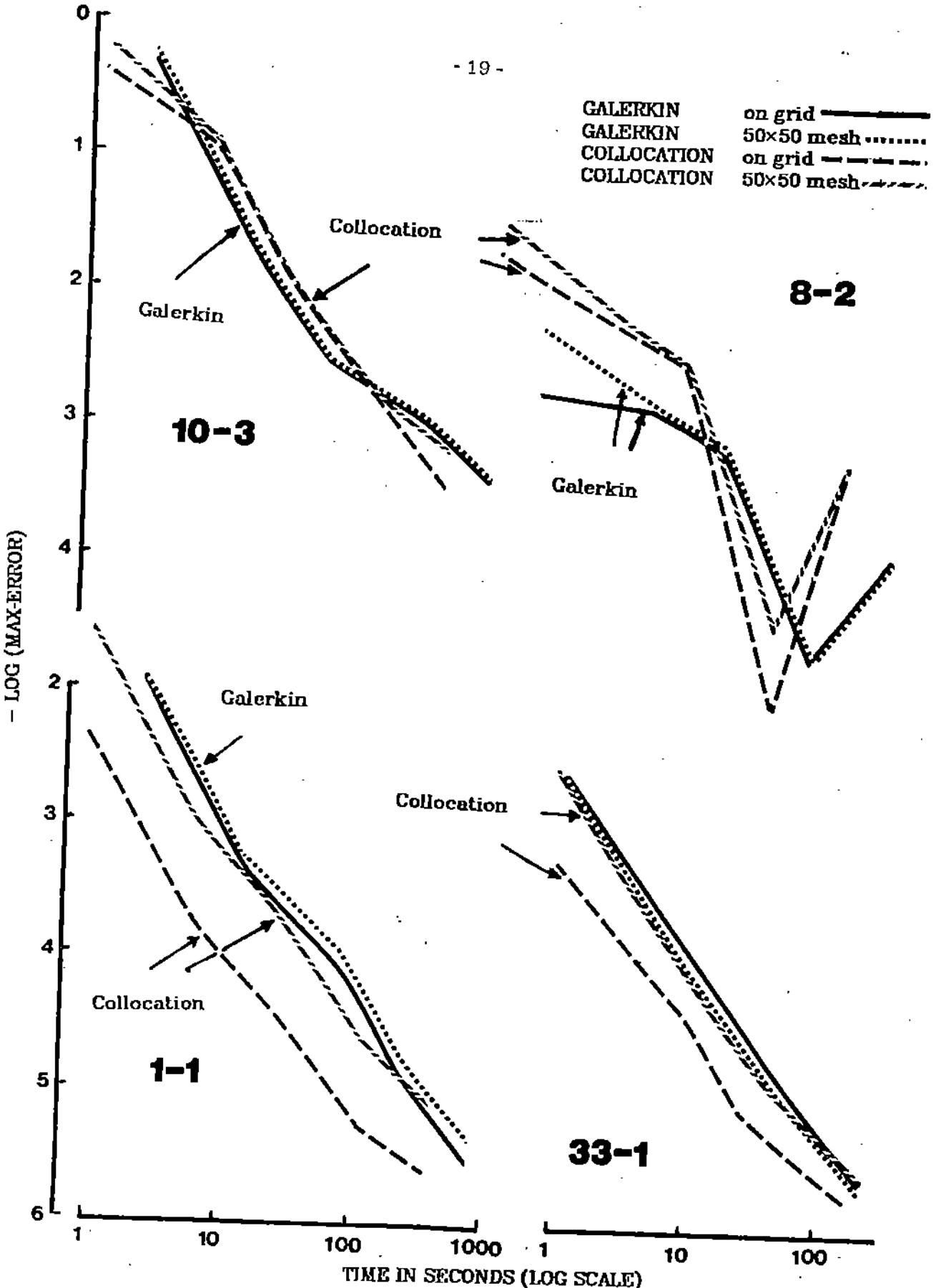



Figure 7. Typical performance profiles of INTERIOR COLLOCATION and SPLINE GALERKIN (DEGREE = 3, SMOOTH = 1) applied to the four problems indicated. The computer time required is plotted on a log scale versus the error measured on the grid and measured on a 50×50 mesh

and on a fixed mesh. Problem 8-2 has Galerkin performance noticeably better except for the case of $n$=20. There is no ready explanation as to why the accuracy in this case is so much better or as to why collocation improves more than Galerkin. For $n$=20 the basis functions have knots along the lines where the third (not second) derivative of the solution has jumps. Problem 10-5 has a solution with a small but sharp peak. The location of this peak relative to the grid lines introduces an erratic behavior into the performance as a function of $n$. One could judge that collocation tends to be better than Galerkin for this problem even though the erratic behavior makes this debatable.

The erratic nature of the performance plots show why one must use statistical techniques to evaluate performance. Figure 7 suggests further that the performance of these two methods are not dramatically different and, even if one is better in some statistical sense, one cannot reliably predict their relative performance in advance. There are enough cases like Problems 1-1 and 33-1 that INTERIOR COLLOCATION is much more likely to outperform SPLINE GALERKIN.

We mentioned earlier that the discretization computations (called assembly by Weiser et al.) is fast compared to the solutions of the linear system. To provide some data for this Table 3 gives the discretization times and solution times for a simple Poisson problem and for Problem 41-3 (the most complex in this set).

*TABLE 3:* Collocation and Galerkin discretization (DIS) and solution (SOL) times for a simple problem (4-1) and a complex problem (41-3).

| | Problem 4-1 Time for | | | | Problem 41-3 Time for | | | |
|---|---|---|---|---|---|---|---|---|
| | Collocation | | Galerkin | | Collocation | | Galerkin | |
| $n$ | DIS | SOL | DIS | SOL | DIS | SOL | DIS | SOL |
| 4 | 0.1 | 0.5 | 0.75 | 1.2 | 1.1 | 0.5 | 3.0 | 1.3 |
| 8 | 0.3 | 4.6 | 2.6 | 9.2 | 4.1 | 4.6 | 10.9 | 9.2 |
| 12 | 0.6 | 17.4 | 6.1 | 35.0 | 9.0 | 17.4 | 25.5 | 36.0 |
| 20 | 1.5 | 103.3 | 18.4 | 212.9 | 24.8 | 103.0 | 73.2 | 209.5 |
| 28 | 2.6 | 347.7 | 39.2 | 719.3 | 48.5 | 348.2 | 143.4 | 717.1 |

We see from the data for Problem 4-1 that the overhead for HERMITE COLLOCATION is much smaller than SPLINE GALERKIN (DEGREE = 3, SMOOTH = 1). Using SPLINE GALERKIN, for moderate grid sizes (e.g., n= 4 to 12), the discretization time is a significant portion of the total time even for the simplest problems. For more complex operators, the ratio

$$\frac{\text{Galerkin discretization time}}{\text{collocation discretization time}}$$

is about 3. Even for rather fine grids, the discretization time of Galerkin remains significant for moderately complex problems. The discretization time for INTERIOR COLLOCATION is essentially the same as that of HERMITE COLLOCATION as the elimination of uncoupled boundary condition equations is a short computation. For rather complex problems the discretization time will frequently be the dominant factor in the time to solve the problem using SPLINE GALERKIN.

An examination of the actual data for this study allows one to observe the effects of machine round-off. The machine used has about 7 decimal digits of

precision and the discretization error for several of the problems is less than this for the larger values of $n$. The Galerkin equations are symmetric positive definite and thus one expects to see minimal round-off effects in solving these equations by Cholesky factorization as implemented in LINPACK SPD. We, in fact, observe that the round-off effects are minimal. The collocation equations are less structured so one might expect round-off effects to be serious when $n=28$ (3300 equations). This is not the case provided Gauss elimination *with scaled partial pivoting* is used. This aspect of the computations is studied further in [Dyksen and Rice,1982]. The data of this study show no significant (or even suggestive) advantage for either method as far as sensitivity to round-off is concerned.

## 6. DISCUSSION OF RESULTS AND CONCLUSIONS

The general question addressed in this study is: *Is the Galerkin method better than collocation*? This question is too general and vague so the following much more specific question is actually addressed: *How do the programs* INTERIOR COLLOCATION + BAND GE *and* SPLINE GALERKIN (DEGREE = 3, SMOOTH = 1) +LINPACK SPD *compare for well behaved linear elliptic problems in two variables*? We believe that these four programs are high quality implementations of the methods upon which they are based and that our conclusions are valid for comparing the Galerkin and collocation discretization methods using direct elimination and Hermite bi-cubic basis functions.

We first list the conclusions which are indisputable or established with high statistical significance; they are listed in decreasing order of confidence

1. The same amount of memory is needed by the two programs.
2. Both methods are reasonably insensitive to round-off error effects.
3. Collocation requires much less computer time than Galerkin to do the discretization.
4. Collocation requires less computer time to achieve 3 digits of accuracy at the grid points (99% confidence).
5. The slope of computer time versus error at the grid points is better for collocation than Galerkin (90% confidence).
6. For a given value of $n$, the error in the Galerkin discretization is smaller than that of the collocation discretization.

Further, we note that there is no difference significant at the 80% level or higher between the two methods in the following comparisons.

7. Computer time versus error at a fixed $20\times20$ mesh (slope or achievement of 3 digits)

We believe the evaluation of the performances of these two methods should be made with the assumption that the problem has homogeneous boundary conditions. It is easy to homogenize the boundary conditions (it is done automatically within ELLPACK if so specified) and benefit is more than the cost for both methods. The program SPLINE GALERKIN does not take advantage of homogeneous boundary conditions but we believe (based on some analysis and experiments) that the possible improvement would not change the performance evaluation results obtained here. The fact that the collocation method sometimes achieves extra accuracy at the grid points can only be viewed as an advantage for it; many applications do not require the results on a very fine grid. This situation has led some authors, for example, [Schultz, 1972] to define the numerical solution of a problem to be a table of values on the grid even if the numerical method produces a function which can be evaluated at any point.

Overall, we conclude that for moderate accuracy the collocation discretization is more efficient than the Galerkin discretization when using Hermite bicubic and coupled with direct elimination methods and applied to smooth, linear elliptic problems. The collocation method has an advantage that is irrelevant to the specific study of this paper but which is significant in a larger context. This method is simple to understand and easily generalizes to problems which are not self-adjoint or which involve more complicated boundary conditions. Its generalization to problems with non-rectangular domains is also easier than for most methods. On the other hand, mathematical analysis of the collocation method is more difficult than that of the Galerkin method.

Note that it is almost certain that it is a poor tactic to solve the linear equations from these discretizations by a direct method, see [Rice, 1981a]. Iteration methods will work for both discretizations and these will be more efficient for larger problems (more than a few hundred unknowns). There is, however, no definitive data on the efficiency of iteration methods for the collocation equations but we suspect that the situation here is similar to that for direct methods, namely the efficiencies are quite comparable for the collocation and Galerkin equations.

Our study of the techniques to measure the error in the numerical solution results in the following conclusions.

1. The error measured at the grid points is reasonably close to the maximum error (except for very coarse grids).
2. There is a special behavior of the error at the grid points compared to that in a fixed mesh. The nature of this behavior is not well understood for either the collocation or Galerkin discretizations. There is a definite (statistically significant, but not uniform) tendency for the collocation error to be smaller at the grid points than at some other fixed mesh. This tendency is strongest for homogeneous boundary conditions.
3. The special behavior mentioned in 2 can affect the performance rankings of closely competitive methods.
4. There is a better way to measure the maximum error than to use a large fixed mesh. The way proposed in Section 4 is both more efficient and more reliable for well behaved problems.
5. There is no completely reliable general method to measure the maximum error for singular problems.

Finally, we observe that while the collocation discretization is superior to Galerkin in the present context, the difference between them is small compared to differences arising from other sources. Recall from approximation theory that it has long been recognized that the choice of norm (which corresponds here to the choice between collocation or Galerkin in the discretization) is secondary to the choice of basis functions. We believe this also to be the case for numerical methods for elliptic problems.

## REFERENCES

[1] R.F. Boisvert, E.N. Houstis and J.R. Rice, A system for performance evaluation of partial differential equation software, IEEE Trans. Software Engineering, 5 (1979) pp 418-425.

[2] H. Crowder, R.S. Bembo and J.M. Mulvey, On reporting computational experiments with mathematical software, ACM Trans. Math. Software, 5 (1979) pp 193-203.

[3] W.R. Dyksen and J.R. Rice, On Gauss elimination for the linear systems arising from the collocation method. CSD-TR XXX, November 1982.

[4] M. Hollander and D.A. Wolfe, *Nonparametric Statistical Methods* John Wiley (1975).

[5] E.N. Houstis and J.R. Rice, An experimental design for the computational evaluation of elliptic partial differential equation solvers, in *Production and Assessment of Numerical Software* (M. Hennell and L. Delves, ed.) Academic Press (1980) pp 57-66.

[6] E.N. Houstis, R.E. Lynch, T.S. Papatheodorou and J.R. Rice, Evaluation of numerical methods for elliptic partial differential equations, J. Comp. Physics, 27 (1978) pp 323-350.

[7] J.R. Rice, ELLPACK: Progress and plans, in *Elliptic Problem Solvers* (M. Schultz, ed.) Academic Press (1981) pp 135-162.

[8] J.R. Rice, E.N. Houstis and W.R. Dyksen, A population of linear, second order, elliptic partial differential equations on rectangular domains, Math. Comp. 36 (1981) pp 475-484.

[9] J.R. Rice, Machine and system effects on the performance of partial differential equations software, Proc. 10th IMACS World Congress, 1 (1982) pp 446-448.

[10] J.R. Rice, Methodology for the algorithm selection problem, in *Performance Evaluation of Numerical Software* (L. Fosdick, ed.), North-Holland (1979) pp 301-307.

[11] J.R. Rice, On the performance of 13 methods to solve the Galerkin method equations, CSD-TR 369, May (1981a).

[12] M. Schultz, The computational complexity of elliptic partial differential equations, "Complexity of Computer Computation", (R.E. Miller and J.W. Thatcher, eds), Plenum Press, New York, 1972.

[13] A. Weiser, S.C. Eisenstat and M.H. Schultz, On solving elliptic problems to moderate accuracy, SIAM J. Numer. Anal., 17 (1980) pp 908-929.

## APPENDIX 1: THE PERFORMANCE DATA

This appendix gives the data generated for this study plus some other data that might be of interest. The data is given for collocation first followed by the same data for Galerkin. Specific definitions of the data items are

| | |
|---|---|
| n | Number of x and y grid squares |
| N | Number of linear equations to solve |
| Err-Grid | Maximum error at the grid points (normalized by the size of U) |
| Err-20x20 | Maximum error at a fixed 20x20 mesh (normalized by the size of U) |
| Time D | Discretization time for INTERIOR COLLOCATION or SPLINE GALERKIN |
| Time I | Indexing time for AS IS (always negligible) |
| Time S | Solution time for BAND GE or LINPACK SPD BAND |
| Time T | TimeD + Time I + Time S |

Collocation Performance Data

Problem 1-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 5.5e-04 | 1.5e-03 | 3.5e-04 | .78 | .22 | .02 | .55 |
| 8 | 256 | 3.6e-05 | 9.9e-05 | 2.8e-05 | 5.12 | .57 | .03 | 4.52 |
| 12 | 576 | 7.0e-06 | 1.9e-05 | 5.6e-06 | 18.87 | 1.33 | .02 | 17.52 |
| 20 | 1600 | 1.8e-06 | 3.2e-06 | 1.2e-06 | 106.60 | 3.30 | .05 | 103.25 |
| 28 | 3136 | 9.8e-06 | 1.0e-05 | 9.4e-06 | 354.62 | 6.53 | .05 | 348.03 |

Problem 3-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 2.6e-03 | 8.9e-03 | 1.6e-03 | .72 | .17 | .03 | .52 |
| 8 | 256 | 1.2e-03 | 2.8e-03 | 9.8e-04 | 5.13 | .58 | .02 | 4.53 |
| 12 | 576 | 7.0e-04 | 1.6e-03 | 6.2e-04 | 18.58 | 1.25 | .03 | 17.30 |
| 20 | 1600 | 3.5e-04 | 3.5e-04 | 3.3e-04 | 106.53 | 3.27 | .03 | 103.23 |
| 28 | 3136 | 2.2e-04 | 2.4e-04 | 2.0e-04 | 352.83 | 6.50 | .05 | 346.28 |

Problem 4-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 2.7e-04 | 1.0e-03 | 8.8e-05 | .65 | .10 | .02 | .53 |
| 8 | 256 | 1.5e-05 | 5.9e-05 | 7.0e-06 | 4.93 | .33 | .03 | 4.57 |
| 12 | 576 | 3.0e-06 | 1.3e-05 | 1.4e-06 | 18.00 | .57 | .03 | 17.40 |
| 20 | 1600 | 6.0e-06 | 4.5e-06 | 2.9e-06 | 104.77 | 1.47 | .03 | 103.27 |
| 28 | 3136 | 1.1e-05 | 1.1e-05 | 6.1e-06 | 350.42 | 2.63 | .05 | 347.73 |

Problem 5-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 3.1e-03 | 4.9e-03 | 1.5e-03 | .70 | .12 | .02 | .57 |
| 8 | 256 | 1.9e-04 | 4.0e-04 | 1.2e-04 | 4.95 | .25 | .03 | 4.67 |
| 12 | 576 | 3.7e-05 | 7.5e-05 | 2.5e-05 | 18.23 | .55 | .03 | 17.65 |
| 20 | 1600 | 4.5e-06 | 8.1e-06 | 3.3e-06 | 106.20 | 1.47 | .05 | 104.68 |
| 28 | 3136 | 2.3e-06 | 3.8e-06 | 1.8e-06 | 355.40 | 2.88 | .05 | 352.47 |

Problem 5-4

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 3.1e-03 | 4.7e-03 | 1.5e-03 | .73 | .15 | .03 | .55 |
| 8 | 256 | 1.9e-04 | 4.0e-04 | 1.2e-04 | 4.95 | .32 | .02 | 4.62 |
| 12 | 576 | 3.8e-05 | 7.3e-05 | 2.5e-05 | 18.33 | .62 | .03 | 17.68 |
| 20 | 1600 | 3.5e-06 | 9.2e-06 | 2.7e-06 | 106.03 | 1.53 | .03 | 104.47 |
| 28 | 3136 | 2.7e-06 | 4.5e-06 | 2.1e-06 | 355.05 | 2.80 | .05 | 352.20 |

Problem 6-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 5.0e-02 | 1.1e-01 | 2.8e-02 | .73 | .22 | .02 | .50 |
| 8 | 256 | 4.0e-03 | 5.9e-03 | 2.6e-03 | 5.30 | .70 | .03 | 4.57 |
| 12 | 576 | 7.5e-04 | 8.5e-04 | 5.7e-04 | 18.97 | 1.48 | .03 | 17.45 |
| 20 | 1600 | 9.7e-05 | 1.2e-04 | 7.2e-05 | 107.90 | 3.98 | .03 | 103.88 |
| 28 | 3136 | 2.5e-05 | 4.0e-05 | 1.9e-05 | 354.52 | 7.47 | .05 | 347.00 |

Collocation Performance Data

Problem 8-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 2.0e-02 | 3.3e-02 | 8.6e-03 | .85 | .22 | .03 | .60 |
| 8 | 256 | 6.6e-03 | 7.7e-03 | 3.6e-03 | 5.30 | .55 | .03 | 4.72 |
| 12 | 576 | 3.3e-03 | 3.2e-03 | 1.6e-03 | 18.45 | .92 | .03 | 17.50 |
| 20 | 1600 | 3.3e-06 | 1.6e-05 | 2.7e-06 | 110.78 | 2.43 | .03 | 108.32 |
| 28 | 3136 | 6.2e-04 | 6.2e-04 | 3.4e-04 | 363.03 | 4.75 | .05 | 358.23 |

Problem 9-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 2.7e-03 | 2.1e-02 | 1.3e-03 | .82 | .20 | .05 | .57 |
| 8 | 256 | 2.2e-04 | 2.6e-03 | 1.1e-04 | 5.45 | .55 | .03 | 4.87 |
| 12 | 576 | 4.6e-05 | 6.3e-04 | 2.2e-05 | 19.07 | 1.07 | .03 | 17.97 |
| 20 | 1600 | 5.9e-06 | 1.1e-05 | 2.9e-06 | 107.68 | 2.63 | .03 | 105.02 |
| 28 | 3136 | 1.4e-06 | 2.4e-05 | 7.9e-07 | 369.33 | 5.13 | .05 | 364.15 |

Problem 10-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 3.3e-01 | 3.5e-01 | 1.4e-02 | .72 | .12 | .03 | .57 |
| 8 | 256 | 1.9e-02 | 9.6e-03 | 2.8e-04 | 4.93 | .38 | .03 | 4.52 |
| 12 | 576 | 3.0e-03 | 1.8e-03 | 3.4e-05 | 18.15 | .67 | .03 | 17.45 |
| 20 | 1600 | 3.6e-04 | 6.6e-04 | 4.2e-06 | 105.30 | 1.70 | .05 | 103.55 |
| 28 | 3136 | 9.0e-05 | 9.0e-05 | 1.1e-06 | 349.47 | 3.25 | .05 | 346.17 |

Problem 10-3

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 3.5e-01 | 5.7e-01 | 2.3e-02 | .73 | .17 | .02 | .55 |
| 8 | 256 | 1.2e-01 | 1.0e-01 | 2.3e-03 | 4.93 | .33 | .03 | 4.57 |
| 12 | 576 | 1.3e-02 | 4.4e-03 | 1.4e-04 | 18.08 | .65 | .03 | 17.40 |
| 20 | 1600 | 1.4e-03 | 2.0e-03 | 1.2e-05 | 104.78 | 1.57 | .03 | 103.18 |
| 28 | 3136 | 3.5e-04 | 3.3e-04 | 3.0e-06 | 349.57 | 3.08 | .05 | 346.43 |

Problem 11-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 4.5e-02 | 9.0e-02 | 2.1e-02 | .73 | .15 | .02 | .57 |
| 8 | 256 | 3.5e-03 | 9.8e-03 | 1.7e-03 | 5.28 | .42 | .02 | 4.85 |
| 12 | 576 | 7.5e-04 | 1.6e-03 | 3.8e-04 | 18.18 | .83 | .03 | 17.32 |
| 20 | 1600 | 9.7e-05 | 1.0e-04 | 5.2e-05 | 108.70 | 2.10 | .05 | 106.55 |
| 28 | 3136 | 2.5e-05 | 6.4e-05 | 1.2e-05 | 356.00 | 4.05 | .05 | 351.90 |

Problem 17-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 2.2e-02 | 2.4e-01 | 1.9e-02 | .73 | .18 | .02 | .53 |
| 8 | 256 | 8.3e-02 | 8.2e-02 | 7.5e-02 | 5.13 | .45 | .02 | 4.67 |
| 12 | 576 | 1.3e-02 | 1.3e-02 | 8.3e-03 | 18.68 | .93 | .02 | 17.73 |
| 20 | 1600 | 3.2e-04 | 5.9e-04 | 2.0e-04 | 106.85 | 2.27 | .05 | 104.53 |
| 28 | 3136 | 1.1e-04 | 2.7e-04 | 7.2e-05 | 356.52 | 4.15 | .05 | 352.32 |

Collocation Performance Data

Problem 22-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 7.8e-06 | 9.5e-05 | 1.0e-04 | .85 | .35 | .02 | .48 |
| 8 | 256 | 6.7e-07 | 5.9e-06 | 9.1e-06 | 5.98 | .93 | .03 | 5.02 |
| 12 | 576 | 7.0e-07 | 1.1e-06 | 8.3e-06 | 20.07 | 2.17 | .03 | 17.87 |
| 20 | 1600 | 3.6e-06 | 3.5e-06 | 5.2e-05 | 112.90 | 5.78 | .03 | 107.08 |
| 28 | 3136 | 1.2e-05 | 1.2e-05 | 2.0e-04 | 369.50 | 11.05 | .07 | 358.38 |

Problem 33-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 4.9e-03 | 2.7e-02 | 1.6e-02 | .82 | .22 | .03 | .57 |
| 8 | 256 | 2.1e-04 | 1.1e-03 | 1.4e-03 | 5.25 | .53 | .03 | 4.68 |
| 12 | 576 | 4.4e-05 | 2.3e-04 | 3.0e-04 | 20.00 | .98 | .02 | 19.00 |
| 20 | 1600 | 5.6e-06 | 3.0e-05 | 4.0e-05 | 111.10 | 2.52 | .03 | 108.55 |
| 28 | 3136 | 2.6e-06 | 7.9e-06 | 1.2e-05 | 364.97 | 5.03 | .08 | 359.85 |

Problem 41-3

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 5.7e-04 | 4.6e-03 | 3.6e-04 | 1.62 | 1.05 | .02 | .55 |
| 8 | 256 | 6.2e-05 | 1.2e-03 | 3.6e-05 | 8.70 | 4.05 | .03 | 4.62 |
| 12 | 576 | 2.5e-05 | 3.6e-04 | 1.0e-05 | 26.43 | 9.00 | .03 | 17.40 |
| 20 | 1600 | 2.1e-05 | 1.7e-05 | 4.4e-06 | 127.82 | 24.80 | .03 | 102.98 |
| 28 | 3136 | 2.4e-05 | 1.7e-05 | 4.8e-06 | 396.78 | 48.52 | .05 | 348.22 |

Problem 47-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 3.0e-05 | 1.7e-04 | 2.2e-05 | .85 | .17 | .03 | .65 |
| 8 | 256 | 7.4e-06 | 3.3e-05 | 5.3e-06 | 5.27 | .43 | .02 | 4.82 |
| 12 | 576 | 3.4e-06 | 1.2e-05 | 2.3e-06 | 19.40 | .82 | .03 | 18.55 |
| 20 | 1600 | 1.2e-06 | 1.2e-06 | 8.0e-07 | 110.23 | 2.03 | .05 | 108.15 |
| 28 | 3136 | 2.0e-06 | 2.0e-06 | 1.1e-06 | 350.52 | 3.70 | .05 | 346.77 |

Problem 50-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 3.4e-03 | 2.4e-02 | 2.8e-03 | .67 | .15 | .02 | .50 |
| 8 | 256 | 2.3e-04 | 2.0e-03 | 2.0e-04 | 4.88 | .28 | .02 | 4.58 |
| 12 | 576 | 3.7e-05 | 5.2e-04 | 4.2e-05 | 18.77 | .58 | .05 | 18.13 |
| 20 | 1600 | 3.9e-06 | 6.9e-05 | 5.4e-06 | 105.85 | 1.50 | .05 | 104.30 |
| 28 | 3136 | 7.7e-06 | 1.9e-05 | 1.0e-05 | 345.82 | 2.23 | .05 | 343.53 |

Problem 54-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 64 | 2.7e-01 | 5.0e-01 | 2.6e-01 | .82 | .23 | .02 | .57 |
| 8 | 256 | 8.1e-02 | 1.0e-01 | 5.4e-02 | 5.10 | .48 | .02 | 4.60 |
| 12 | 576 | 1.0e-02 | 1.3e-02 | 5.4e-03 | 19.08 | 1.12 | .03 | 17.93 |
| 20 | 1600 | 4.8e-04 | 1.0e-03 | 2.3e-04 | 109.40 | 2.97 | .03 | 106.40 |
| 28 | 3136 | 9.2e-05 | 2.0e-04 | 5.6e-05 | 357.37 | 5.42 | .05 | 351.90 |

Number of ELLPACK runs = 90

Total CPU hours = 2.46

Galerkin Performance Data

Problem 1-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 7.9e-04 | 8.0e-04 | 6.3e-04 | 2.03 | .85 | .02 | 1.17 |
| 8 | 324 | 6.7e-05 | 6.6e-05 | 5.4e-05 | 12.73 | 3.55 | .02 | 9.17 |
| 12 | 676 | 1.4e-05 | 1.1e-05 | 1.2e-05 | 43.22 | 7.35 | .03 | 35.83 |
| 20 | 1764 | 2.3e-06 | 1.1e-06 | 1.9e-06 | 233.77 | 22.60 | .03 | 211.13 |
| 28 | 3364 | 1.8e-06 | 1.4e-06 | 1.2e-06 | 783.15 | 49.18 | .05 | 733.92 |

Problem 3-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 3.3e-03 | 6.0e-03 | 1.9e-03 | 2.13 | .88 | .03 | 1.22 |
| 8 | 324 | 6.5e-04 | 2.1e-03 | 3.8e-04 | 12.80 | 3.57 | .03 | 9.20 |
| 12 | 676 | 2.3e-04 | 6.5e-04 | 2.0e-04 | 41.83 | 7.42 | .03 | 34.38 |
| 20 | 1764 | 1.0e-04 | 1.1e-04 | 1.0e-04 | 235.35 | 23.45 | .05 | 211.85 |
| 28 | 3364 | 6.4e-05 | 1.3e-04 | 6.8e-05 | 771.98 | 49.40 | .05 | 722.53 |

Problem 4-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 5.9e-04 | 5.5e-04 | 2.6e-04 | 2.00 | .75 | .03 | 1.22 |
| 8 | 324 | 4.0e-05 | 3.6e-05 | 2.2e-05 | 11.78 | 2.60 | .03 | 9.15 |
| 12 | 676 | 7.9e-06 | 7.1e-06 | 4.4e-06 | 41.12 | 6.10 | .03 | 34.98 |
| 20 | 1764 | 1.2e-06 | 1.3e-06 | 5.7e-07 | 231.28 | 18.37 | .03 | 212.88 |
| 28 | 3364 | 1.5e-06 | 1.7e-06 | 5.3e-07 | 758.53 | 39.23 | .03 | 719.27 |

Problem 5-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 2.1e-03 | 1.4e-03 | 1.2e-03 | 1.22 | .80 | .02 | .40 |
| 8 | 324 | 2.1e-04 | 1.1e-04 | 1.6e-04 | 11.95 | 2.65 | .03 | 9.27 |
| 12 | 676 | 4.8e-05 | 4.1e-05 | 3.9e-05 | 43.15 | 6.15 | .03 | 36.97 |
| 20 | 1764 | 5.6e-06 | 6.6e-06 | 5.4e-06 | 227.58 | 18.62 | .03 | 208.93 |
| 28 | 3364 | 3.0e-06 | 1.7e-06 | 2.3e-06 | 762.15 | 40.28 | .05 | 721.82 |

Problem 5-4

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 2.1e-03 | 1.4e-03 | 1.2e-03 | 2.02 | .80 | .02 | 1.20 |
| 8 | 324 | 2.1e-04 | 1.1e-04 | 1.6e-04 | 11.72 | 2.67 | .03 | 9.02 |
| 12 | 676 | 4.8e-05 | 4.1e-05 | 3.9e-05 | 43.65 | 6.32 | .05 | 37.28 |
| 20 | 1764 | 8.5e-06 | 4.7e-06 | 6.8e-06 | 230.65 | 18.57 | .03 | 212.05 |
| 28 | 3364 | 2.0e-06 | 2.5e-06 | 1.8e-06 | 752.52 | 39.37 | .05 | 713.10 |

Problem 6-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 2.1e-02 | 3.5e-02 | 1.4e-02 | 2.22 | .93 | .03 | 1.25 |
| 8 | 324 | 2.1e-03 | 1.8e-03 | 1.4e-03 | 12.90 | 3.43 | .02 | 9.45 |
| 12 | 676 | 6.6e-04 | 3.8e-04 | 4.6e-04 | 45.50 | 8.65 | .03 | 36.82 |
| 20 | 1764 | 1.2e-04 | 6.6e-05 | 8.8e-05 | 241.42 | 25.55 | .05 | 215.82 |
| 28 | 3364 | 3.6e-05 | 2.3e-05 | 2.6e-05 | 773.13 | 49.93 | .03 | 723.17 |

Galerkin Performance Data

Problem 8-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 2.3e-03 | 4.6e-03 | 1.2e-03 | 2.10 | .83 | .03 | 1.23 |
| 8 | 324 | 1.5e-03 | 1.9e-03 | 9.8e-04 | 12.47 | 2.95 | .03 | 9.48 |
| 12 | 676 | 6.8e-04 | 6.8e-04 | 4.1e-04 | 43.07 | 6.80 | .03 | 36.23 |
| 20 | 1764 | 2.0e-05 | 1.1e-05 | 1.2e-05 | 232.82 | 21.42 | .03 | 211.37 |
| 28 | 3364 | 1.2e-04 | 1.4e-04 | 7.4e-05 | 841.95 | 41.75 | .07 | 800.13 |

Problem 9-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 6.7e-03 | 6.7e-03 | 5.2e-03 | 2.18 | .95 | .03 | 1.20 |
| 8 | 324 | 8.5e-04 | 7.7e-04 | 5.3e-04 | 12.33 | 3.07 | .02 | 9.25 |
| 12 | 676 | 2.4e-04 | 2.0e-04 | 1.3e-04 | 42.68 | 6.78 | .03 | 35.87 |
| 20 | 1764 | 4.1e-05 | 3.7e-05 | 2.0e-05 | 227.17 | 20.32 | .00 | 206.85 |
| 28 | 3364 | 1.2e-05 | 1.1e-05 | 5.7e-06 | 761.18 | 42.73 | .03 | 718.42 |

Problem 10-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 1.4e-01 | 1.3e-01 | 4.1e-03 | 2.00 | .77 | .03 | 1.20 |
| 8 | 324 | 2.3e-03 | 2.7e-03 | 4.2e-05 | 11.75 | 2.68 | .02 | 9.05 |
| 12 | 676 | 1.8e-03 | 1.2e-03 | 2.5e-05 | 41.17 | 6.42 | .03 | 34.72 |
| 20 | 1764 | 4.0e-04 | 3.0e-04 | 5.4e-06 | 229.67 | 18.63 | .05 | 210.98 |
| 28 | 3364 | 1.2e-04 | 1.8e-05 | 1.7e-06 | 754.77 | 40.63 | .05 | 714.08 |

Problem 10-3

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 1.4e-01 | 1.6e-01 | 4.9e-03 | 2.02 | .77 | .03 | 1.22 |
| 8 | 324 | 1.8e-02 | 1.9e-02 | 3.6e-04 | 12.10 | 2.80 | .02 | 9.28 |
| 12 | 676 | 2.6e-03 | 3.3e-03 | 3.3e-05 | 42.15 | 6.33 | .03 | 35.78 |
| 20 | 1764 | 1.1e-03 | 7.5e-04 | 1.1e-05 | 228.63 | 18.60 | .05 | 209.98 |
| 28 | 3364 | 3.9e-04 | 4.4e-05 | 3.8e-06 | 755.33 | 40.62 | .03 | 714.68 |

Problem 11-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 2.6e-02 | 2.6e-02 | 1.8e-02 | 2.12 | .80 | .03 | 1.28 |
| 8 | 324 | 2.7e-03 | 2.0e-03 | 2.1e-03 | 12.15 | 2.88 | .03 | 9.23 |
| 12 | 676 | 9.0e-04 | 6.1e-04 | 5.4e-04 | 42.70 | 6.93 | .03 | 35.73 |
| 20 | 1764 | 1.5e-04 | 1.3e-04 | 8.4e-05 | 228.65 | 19.12 | .03 | 209.50 |
| 28 | 3364 | 4.3e-05 | 3.4e-05 | 2.3e-05 | 755.55 | 41.00 | .03 | 714.52 |

Problem 17-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 1.2e-01 | 1.7e-01 | 1.3e-01 | 2.05 | .80 | .03 | 1.22 |
| 8 | 324 | 1.3e-02 | 1.3e-02 | 1.3e-02 | 12.03 | 2.80 | .02 | 9.22 |
| 12 | 676 | 1.5e-03 | 1.1e-03 | 1.2e-03 | 44.52 | 6.48 | .03 | 38.00 |
| 20 | 1764 | 2.5e-04 | 2.6e-04 | 1.9e-04 | 251.10 | 20.70 | .05 | 230.35 |
| 28 | 3364 | 8.6e-05 | 9.2e-05 | 7.0e-05 | 760.65 | 42.03 | .07 | 718.55 |

Galerkin Performance Data

Problem 22-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 4.1e-05 | 4.1e-05 | 8.1e-04 | 1.85 | .55 | .02 | 1.28 |
| 8 | 324 | 3.1e-06 | 7.1e-06 | 6.2e-05 | 15.00 | 4.45 | .03 | 10.52 |
| 12 | 676 | 6.7e-07 | 1.0e-06 | 8.3e-06 | 43.83 | 8.93 | .03 | 34.87 |
| 20 | 1764 | 4.5e-07 | 4.5e-07 | 3.4e-06 | 232.77 | 26.58 | .03 | 206.15 |
| 28 | 3364 | 1.2e-06 | 1.3e-06 | 1.4e-05 | 765.73 | 55.73 | .05 | 709.95 |

Problem 33-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 1.2e-02 | 1.1e-02 | 4.7e-02 | 2.18 | .92 | .02 | 1.25 |
| 8 | 324 | 5.8e-04 | 5.2e-04 | 2.9e-03 | 13.08 | 3.12 | .02 | 9.95 |
| 12 | 676 | 1.2e-04 | 1.1e-04 | 5.5e-04 | 46.90 | 7.40 | .05 | 39.45 |
| 20 | 1764 | 1.5e-05 | 1.3e-05 | 7.1e-05 | 243.70 | 21.33 | .03 | 222.33 |
| 28 | 3364 | 4.2e-06 | 3.5e-06 | 1.7e-05 | 768.25 | 44.00 | .08 | 724.17 |

Problem 41-3

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 5.7e-04 | 1.5e-03 | 4.7e-04 | 4.27 | 2.98 | .02 | 1.27 |
| 8 | 324 | 1.1e-04 | 3.9e-04 | 5.8e-05 | 20.12 | 10.92 | .02 | 9.18 |
| 12 | 676 | 4.9e-05 | 7.9e-05 | 1.8e-05 | 61.47 | 25.48 | .03 | 35.95 |
| 20 | 1764 | 2.1e-05 | 1.7e-05 | 5.5e-06 | 282.72 | 73.17 | .03 | 209.52 |
| 28 | 3364 | 2.4e-05 | 1.7e-05 | 3.9e-06 | 860.62 | 143.42 | .07 | 717.13 |

Problem 47-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 6.7e-05 | 6.7e-05 | 4.8e-05 | 1.95 | .73 | .03 | 1.18 |
| 8 | 324 | 1.2e-05 | 1.3e-05 | 5.2e-06 | 12.87 | 2.97 | .02 | 9.88 |
| 12 | 676 | 4.3e-06 | 4.3e-06 | 1.4e-06 | 44.95 | 6.92 | .03 | 38.00 |
| 20 | 1764 | 1.2e-06 | 1.2e-06 | 2.9e-07 | 240.17 | 19.87 | .03 | 220.27 |
| 28 | 3364 | 5.2e-07 | 5.2e-07 | 1.4e-07 | 752.77 | 42.38 | .07 | 710.32 |

Problem 50-1

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 9.1e-03 | 9.1e-03 | 1.1e-02 | 1.90 | .67 | .03 | 1.20 |
| 8 | 324 | 9.0e-04 | 7.6e-04 | 1.0e-03 | 12.35 | 2.55 | .03 | 9.77 |
| 12 | 676 | 2.3e-04 | 2.0e-04 | 2.5e-04 | 43.10 | 6.05 | .03 | 37.02 |
| 20 | 1764 | 3.5e-05 | 3.0e-05 | 3.7e-05 | 231.83 | 18.58 | .05 | 213.20 |
| 28 | 3364 | 9.6e-06 | 7.4e-06 | 1.0e-05 | 750.83 | 38.28 | .03 | 712.52 |

Problem 54-2

| n | N | Err-Grid | Err-20x20 | Err-L2 | TimeT | TimeD | TimeI | TimeS |
|---|---|---|---|---|---|---|---|---|
| 4 | 100 | 2.2e-01 | 2.7e-01 | 1.3e-01 | 2.05 | .82 | .03 | 1.20 |
| 8 | 324 | 1.5e-02 | 1.6e-02 | 9.3e-03 | 11.92 | 2.95 | .03 | 8.93 |
| 12 | 676 | 1.8e-03 | 1.9e-03 | 1.0e-03 | 41.62 | 6.88 | .02 | 34.72 |
| 20 | 1764 | 2.3e-04 | 2.9e-04 | 1.7e-04 | 232.08 | 20.53 | .03 | 211.52 |
| 28 | 3364 | 8.9e-05 | 4.3e-05 | 6.1e-05 | 760.00 | 45.15 | .07 | 714.78 |

Number of ELLPACK runs = 90

Total CPU hours = 5.34

## APPENDIX 2: THE 18 ELLIPTIC PROBLEMS

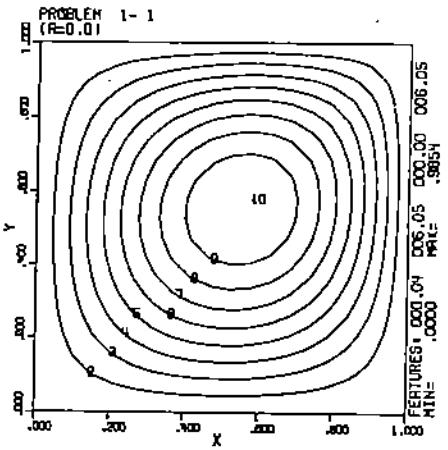
For reference purposes, we include a description of the 18 elliptic problems used in this study along with a contour plot of the true solution. This material is extracted from [J. Rice et al., 1981]

### PROB 1 Artificial [7,12,13]

$$(e^{xy}u_x)_x + (e^{-xy}u_y)_y - u/(1 + x + y) = f$$

DOMAIN unit square
BC $u + \alpha u_N = g$
TRUE $.75e^{xy}\sin(\pi x)\sin(\pi y)$

Operator: Self-adjoint, analytic
Right side: Entire
Boundary conditions: Mixed except for $\alpha = 0$.
Solution: Entire, independent of $\alpha$.
Parameter: $\alpha$ introduces normal derivative into boundary conditions.

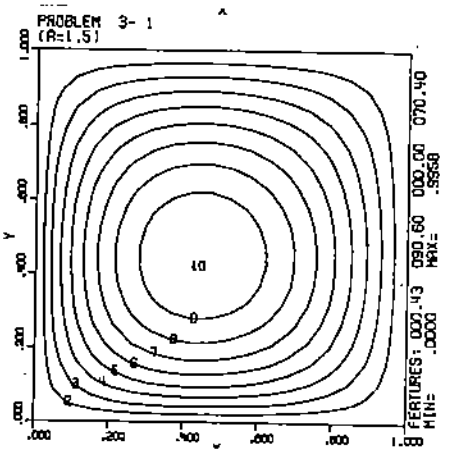



### PROB 3 Artificial [13]

$$u_{xx} + u_{yy} = f$$

DOMAIN unit square
BC $u = 0$
TRUE $c(x^{\alpha/2} - x)(y^{\alpha/2} - y)$, $c = 1/(\alpha^{\alpha/(1-\alpha)} - \alpha^{1/(1-\alpha)})^2$

Operator: Laplace
Right side: singular for $\alpha \leq 3$
Boundary condition: Dirichlet, homogeneous
Parameter: $1 < \alpha \leq 5$ adjusts singularity strength

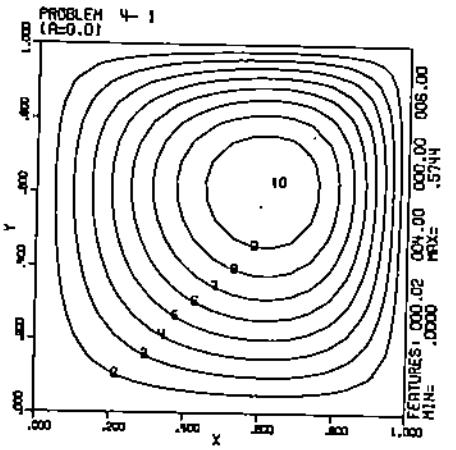



### PROB 4 Artificial [7,12,13]

$$u_{xx} + u_{yy} = 6xy\,e^{x+y}(xy + x + y - 3)$$

DOMAIN unit square
BC $u = 0$ for $x \neq 0$; $u - \alpha(y - y^2)u_x = g$ for $x = 0$
TRUE $3e^{x+y}(x - x^2)(y - y^2)$

Operator: Laplace
Right side: Entire
Boundary conditions: Mixed except for $\alpha = 0$
Solution: Entire, independent of $\alpha$
Parameter: $\alpha$ introduces normal derivative into boundary conditions

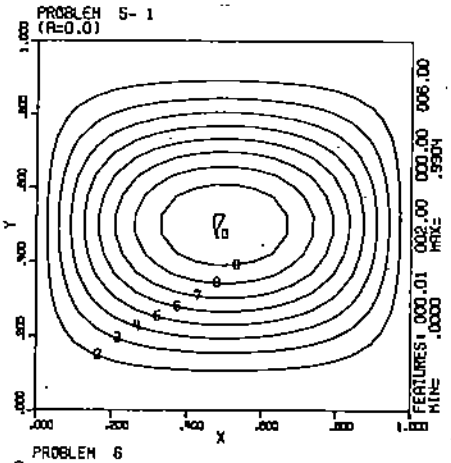


## PROB 5 Artificial [13,14]

$$4u_{xx} + u_{yy} - \alpha u = f$$

**DOMAIN** unit square
**BC** $u = 0$
**TRUE** $2(x^2 - x)(\cos(2\pi y) - 1)$

Operator: Constant coefficient, separable
Right side: Entire
Boundary conditions: Dirichlet, homogeneous
Parameter: $\alpha$ makes operator more singular without affecting solution




## PROB 6 Stratospheric physics [13,14,16]

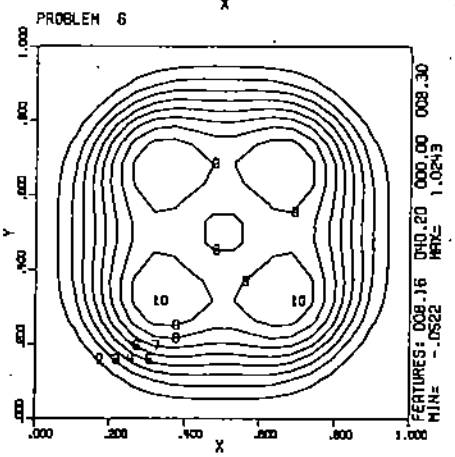
$$u_{xx} + u_{yy} - (100 + \cos(2\pi x) + \sin(3\pi y))u = f$$

**DOMAIN** unit square
**BC** $u = 0$
**TRUE**

$$-0.31(5.4-\cos(4\pi x))\sin(\pi x)(y^2-y)(5.4-\cos(4\pi y))(1/(1+\varphi^4)-.5)$$

$$\varphi = 4(x - .5)^2 + 4(y - .5)^2$$

Operator: Entire, oscillatory, somewhat singular
Right side: Analytic
Boundary conditions: Dirichlet, homogeneous
Parameter: None

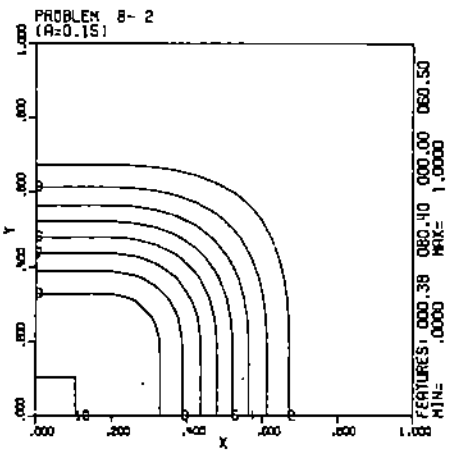



## PROB 8 Artificial [13]

$$u_{xx} + u_{yy} = f$$

**DOMAIN** unit square
**BC** $u = g$
**TRUE** $\varphi(x)\varphi(y)$ where $\varphi(x) = 1$ for $x \leq .5 - \alpha$, $= 0$ for $x \geq .5 + \alpha$ and $\varphi(x)$ is a quintic polynomial for $.5 - \alpha \leq x \leq .5 + \alpha$ so $\varphi$ has two continuous derivatives.

Operator: Laplace
Right side: Just continuous with a right angle ridge.
Boundary conditions: Dirichlet
Solution: Wave front along a right angle joining two regions where it is constant.
Parameter: $\alpha$ adjusts width and sharpness of wave front.

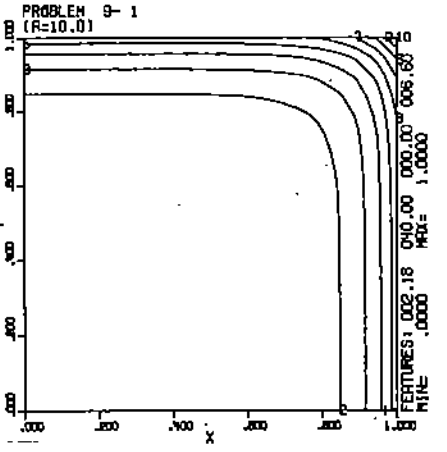



## PROB 9 Artificial [13]

$$u_{xx} + u_{yy} - 100u = .5(\alpha^2 - 100)\cosh(\alpha y)/\cosh\alpha$$

**DOMAIN** unit square
**BC** $u = g$
**TRUE** $.5(\cosh 10x/\cosh 10 + \cosh \alpha y/\cosh \alpha)$

Operator: Helmhotz, constant coefficients, somewhat singular.
Right side: Entire but nearly singular for $\alpha \neq 10$.
Boundary conditions: Dirichlet
Solution: Boundary layer, nearly singular.
Parameter: $\alpha$ adjusts strength of y-side boundary layer.

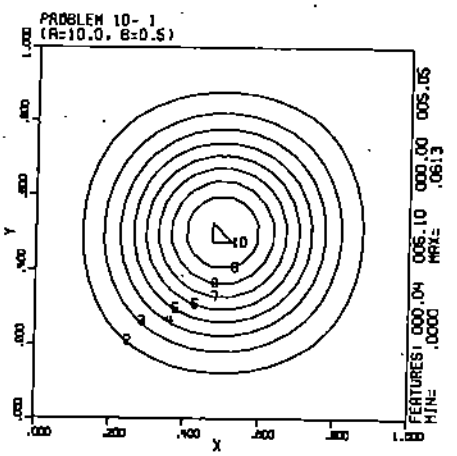


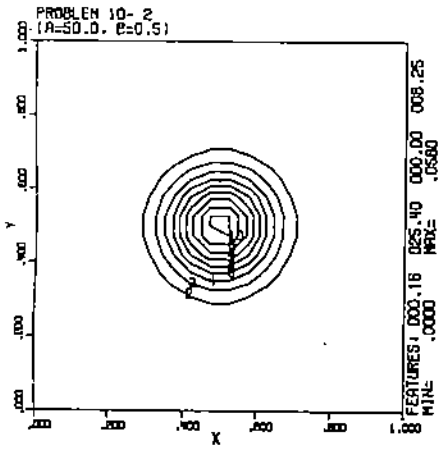
# PROB 10 Artificial [13]

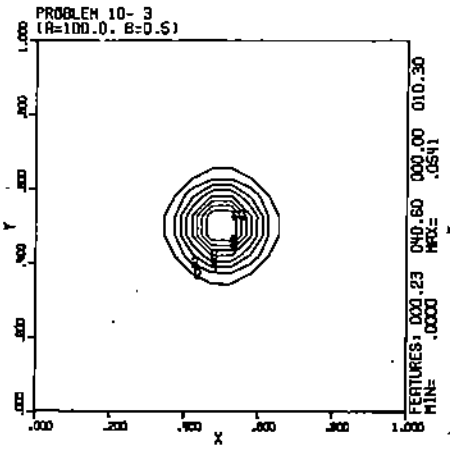
$$u_{xx} + u_{yy} = f$$

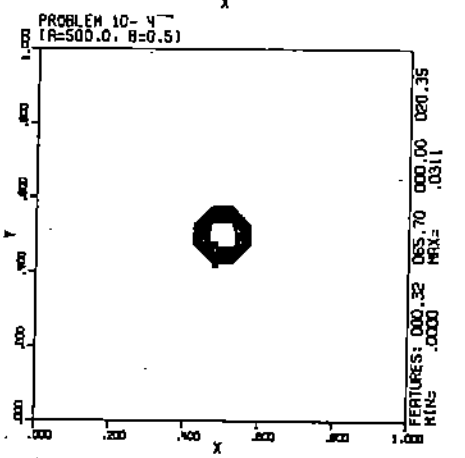
**DOMAIN** unit square
**BC** $u = 0$
**TRUE** $e^{-\alpha[(x-.5)^2+(y-\beta)^2]}(x^2 - x)(y^2 - y)$

Operator: Laplace
Right side: Strongly peaked if $\alpha$ large, but entire.
Boundary condition: Dirichlet, homogeneous
Solution: Strongly peaked for large $\alpha$.
Parameters: $\alpha$ adjusts strength of the peak, $\beta$ moves it in the y-direction.



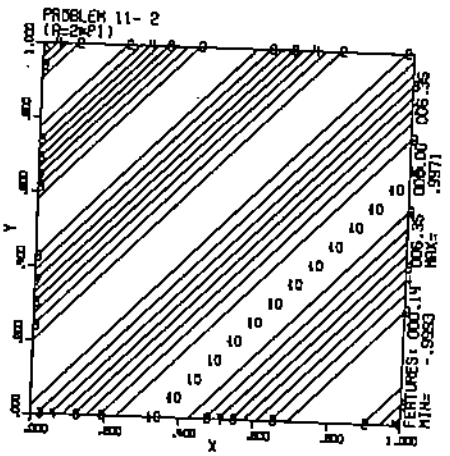




# PROB 11 Artificial

$$u_{xx} + u_{yy} = f$$

**DOMAIN** unit square
**BC** $u = g$
**TRUE** $\sin[\alpha(x - y + 2)^5/(1 + (x - y + 2)^4)]$

Operator: Laplace
Right side: Oscillatory, analytic
Boundary conditions: Dirichlet
Solution: Oscillatory
Parameter: $\alpha$ adjusts frequency of oscillations

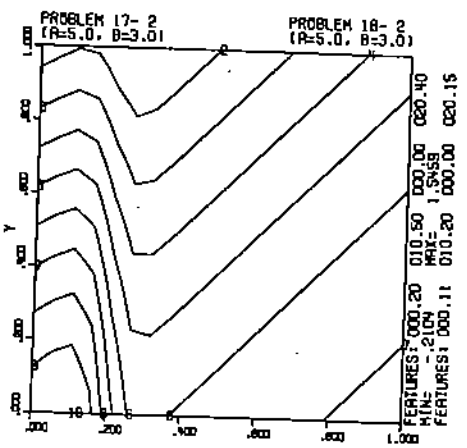



# PROB 17 Artificial

$$u_{xx} + u_{yy} = f$$

**DOMAIN** unit square
**BC** $u = g$
**TRUE** $e^{[y^2+(\alpha(\beta x)^3/(1+(\beta x)^3))^2]} + \sin(x - y + .5)$

Operator: Laplace
Right side: Large values for $x$ near .15
Boundary conditions: Dirichlet
Solution: Sharp wave front near $x = .15$, entire.
Parameters: $\alpha$, $\beta$ adjust the strength and shape of the wave front.

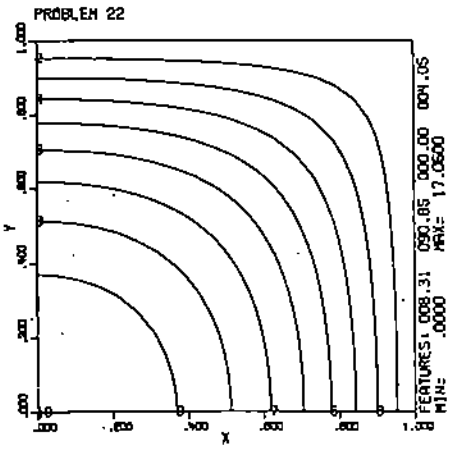


# PROB 22 Elastic-plastic torsion [15]

$w(u_{xx} + u_{yy}) + w_x u_x + w_y u_y = f$, w defined below

**DOMAIN** unit square
**BC** $u = g$
**TRUE** $[17.06 + 3.62(x^2 + y^2)](x^2 - 1)(y^2 - 1)$

Operator: Expanded form of self-adjust problem, discontinuous coefficients. $w = 1/7996$ if $A \leq .0025$
$w = 1/(236 + 19.4/A)$ if $A > .0025$ where $A = \sqrt{T_x^2 + T_y^2}$
Right side: Singular
Boundary conditions: Dirichlet
Solution: T is a quartic polynomial
Parameter: None

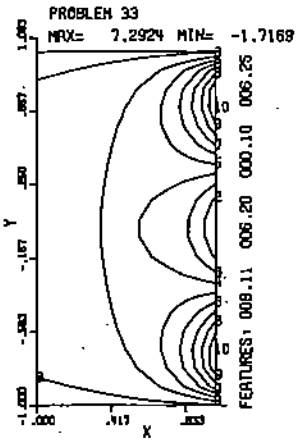



# PROB 33 Torsion on a shaft [5]

$u_{xx} + u_{yy} = f$

**DOMAIN** $[0,1] \times [-1,1]$
**BC** $u = g$
**TRUE** $p = 14 + \sqrt{133}$, $q = 14 - \sqrt{133}$, $r = (7-q)/(r\sqrt{133})$,
$t(y) = 1-y^2$, $C(x) = e^{\sqrt{p}x} - e^{\sqrt{q}x}$, $B(x) = (7-p)r/16C(x)$,
$A(x) = rC(x) + e^{\sqrt{q}x}$, $TRUE = t(y)[A(x) + t(y)B(x)]$

Operator: Laplace
Right side: Entire
Boundary conditions: Dirichlet
Solution: Entire



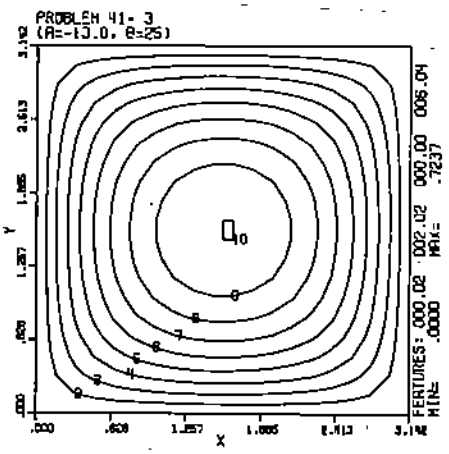

# PROB 41 Artificial [20]

$u_{xx} + u_{yy} + \alpha u = f$

**DOMAIN** $[0,\pi] \times [0,\pi]$
**BC** $u = 0$
**TRUE** approximate solution accuracy depends on $\beta$

$$\frac{x(\pi-x)}{2} - \frac{4}{\pi}\sum_{k=1}^{\beta} \frac{\sin[(2k-1)x]\cosh[(2k-1)(y-\pi/2)]}{(2k-1)^3\cosh[(2k-1)\pi/2]}$$

Operator: Helmholtz
Right side: Series for function with singularities.
Boundary conditions: Dirichlet, homogeneous.
Solution: Infinite series converging like $1/k^3$. The solution has derivative singularities.
Parameters: $\alpha$ adjust u term, possibly makes operator nearly singular. $\beta$ is number of terms in series.

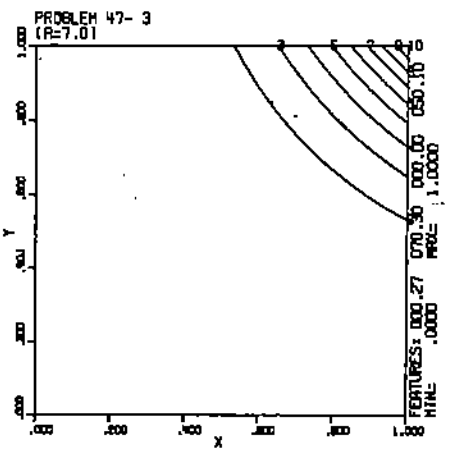



# PROB 47 Artificial

$u_{xx} + u_{yy} = f$

**DOMAIN** unit square
**BC** $u = g$
**TRUE** $(xy)^{\alpha/2}$

Operator: Laplace
Right side: Variable singularities
Boundary conditions: Dirichlet
Solution: Singularity of variable strength.
Parameter: $\alpha$ adjusts singularity strength.



# PROB 50 Artificial [20]

$$u_{xx} + u_{yy} = 0$$

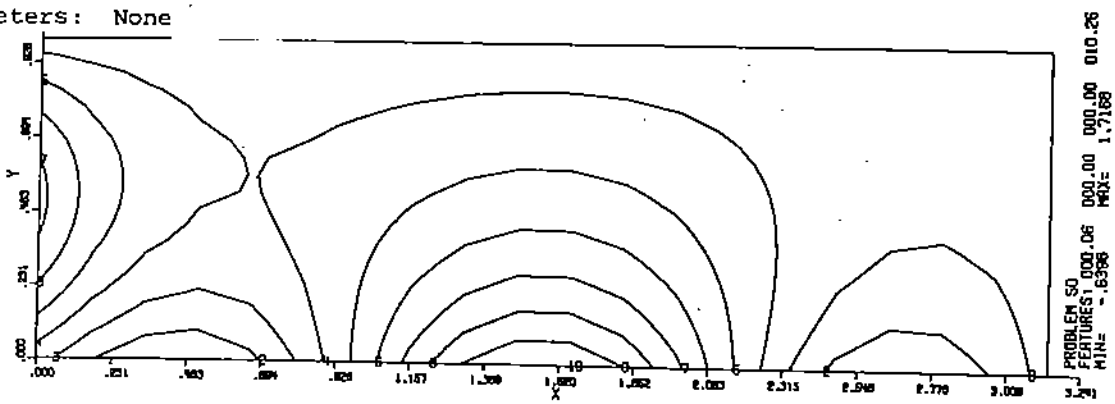
**DOMAIN** $[0,\pi] \times [0,1]$

**BC** $u=3\sin(x)/4-\sin(3x), y=0;\ u=0, x=\pi, y=1;\ u=\sin\pi y,\ x=0$

**TRUE**

$$\frac{3\sinh(1-y)\sin x}{4\sinh 1} - \frac{\sinh 3(1-y)\sin 3x}{\sinh 3} + \frac{\sinh\pi(1-x)\sin\pi y}{\sinh\pi}$$

Operator: Laplace, homogeneous
Right side: Zero
Boundary conditions: Dirichlet
Solution: Entire
Parameters: None




# PROB 54 Artificial

$$(1+x^2)u_{xx}+(1+A^2)u_{yy}+2xu_x+16yAu_y-(1+(8y-x-4)^2)u = f$$

**DOMAIN** unit square $\qquad A(y) = 4y^2 + \alpha$

**BC** $u = g$

**TRUE** $B = \max[0,(3-x/A(y))^3],\ C = \max[0,x-A(y)]$

$D = 0$ if $C < .02$, $D = e^{-B/C}$ if $C \geq .02$

$$u(x,y) = 2.25x(x-A(y))^2(1-D)/(4A(y)^3)+1/(1+(8y-x-4)^2)$$

Operator: Expanded form of self-adjoint operator. Analytic.
Right side: Complicated with possible wild behavior.
Boundary conditions: Dirichlet
Solution: Wildly behaving for $\alpha$ possible, has singularities for $x - 4y^2 = \alpha$ or $4y^2 = -\alpha$.